

METHOD AND SYSTEM FOR RESYNCHRONIZATION OF TOPOLOGY DATABASE IN COMMUNICATION NETWORK HAVING TOPOLOGY STATE ROUTING PROTOCOL

Publication number: JP2002135328 (A)

Publication date: 2002-05-10

Inventor(s): MCALLISTER SHAWN P; RAJSIC CARL +

Applicant(s): ALCATEL CANADA INC +

Classification:

- **international:** **H04L12/24; H04L12/56; H04L12/24; H04L12/56; (IPC1-7): H04L12/56**

- **European:** **H04L12/24C2; H04L12/56C1**

Application number: JP20010270169 20010906

Priority number(s): US20000663793 20000918

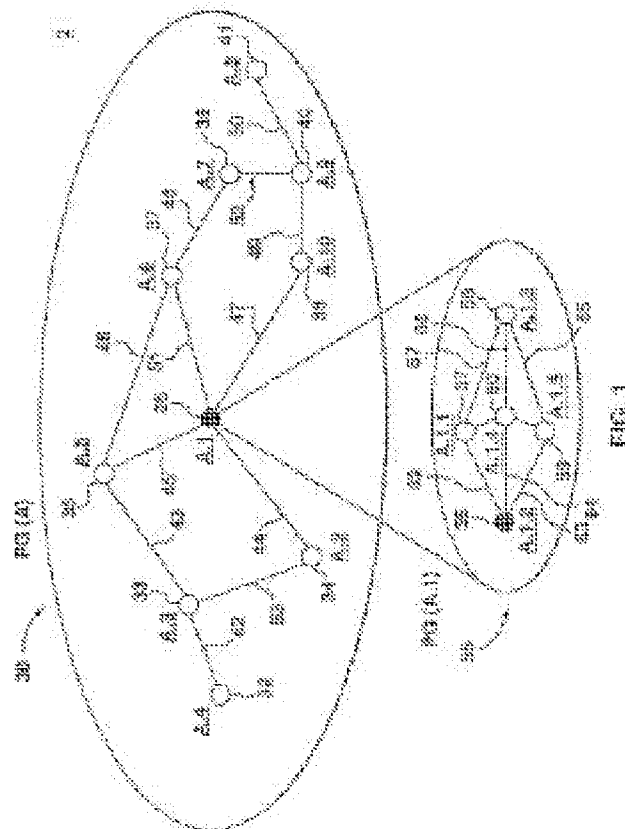
Also published as:

EP1189384 (A2)
EP1189384 (A3)
EP1189384 (B1)
US6876625 (B1)
DE60118143 (T2)

more >>

Abstract of JP 2002135328 (A)

PROBLEM TO BE SOLVED: To provide a method and system for synchronizing topology state information between two network nodes in a communication network. **SOLUTION:** The communication network has a routing protocol that intermittently notifies local state information via a network. Two nodes include a request node that starts a topology state synchronization request and a reply node that receives the request. The reply node provides topology state information not possessed by the request node to the request node. The method includes a selection step of 1st and 2nd nodes for synchronization. The 1st node is prepared for synchronization of the topology state needing stop of the intermittent notice of local state information because the intermittent notice of the local state information relates to the request node and the reply node, and the 2nd node is prepared for synchronization of the topology state maintaining the intermittent notice of the information.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-135328
(P2002-135328A)

(43) 公開日 平成14年5月10日 (2002.5.10)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
H 0 4 L 12/56	1 0 0 4 0 0	H 0 4 L 12/56	1 0 0 Z 5 K 0 3 0 4 0 0 B

審査請求 未請求 請求項の数87 O L 外国語出願 (全 81 頁)

(21) 出願番号	特願2001-270169 (P2001-270169)	(71) 出願人	501279833 アルカテル・カナダ・インコーポレイテツ ド カナダ国、オンタリオ・ケー・2・ケー・ 2・イー・6、カナダ、マーチ・ロード・ 600
(22) 出願日	平成13年9月6日 (2001.9.6)	(72) 発明者	シヤウン・ピー・マクアリスター カナダ国、オンタリオ・ケー・4・エム・ 1・ケー・8、モノテツク、レベル・ドラ イブ・1330
(31) 優先権主張番号	6 6 3 7 9 3	(74) 代理人	100062007 弁理士 川口 義雄 (外5名)
(32) 優先日	平成12年9月18日 (2000.9.18)		
(33) 優先権主張国	米国 (U S)		

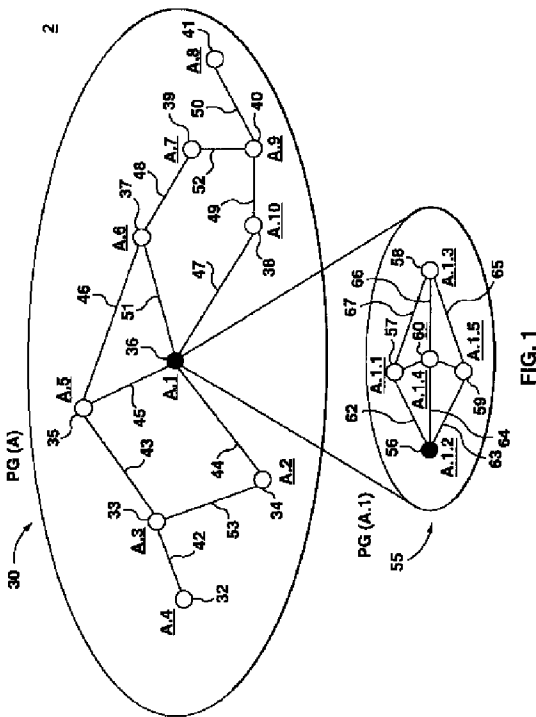
最終頁に続く

(54) 【発明の名称】 トポロジ状態ルーティングプロトコルを有する通信ネットワークにおけるトポロジデータベースの再同期化のための方法および装置

(57) 【要約】

【課題】 通信ネットワークの2つのネットワークノード間におけるトポロジ状態情報を同期化するための方法と装置を提供すること。

【解決手段】 通信ネットワークは、ネットワークを通じて、局所的な状態情報を断続的に通告するルーティングプロトコルを有する。2つのノードは、トポロジ状態同期化要求を開始する要求ノードと、要求を受信する応答ノードを含む。応答ノードは、要求ノードの所有しないトポロジ状態情報を、要求ノードに提供する。該方法は、同期化の第1モードと第2モードの選択ステップを含む。第1モードは、局所的な状態情報の断続的通告が、要求ノードと応答ノードに関係しているので、局所的な状態情報の断続的通告の中止を必要とするトポロジ状態の同期化に備え、第2モードは、同情報の断続的通告を維持するトポロジ状態の同期化に備えている。



【特許請求の範囲】

【請求項1】 通信ネットワークの活動ルーティングエンティティに影響を与える故障から回復するための方法であって、活動ルーティングエンティティは通信ネットワークのネットワークノードと関連付けられており、通信ネットワークはネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルを備え、さらに、前記故障の際に、ネットワークノードのネットワーク接続を活動ルーティングエンティティから転換することができる非活動ルーティングエンティティを備え、

(a) 故障の際に、活動ルーティングエンティティと非活動ルーティングエンティティの間で活動切替えを実行し、ネットワークノードのネットワーク接続が、活動ルーティングエンティティから非活動ルーティングエンティティに転換され、それにより、非活動ルーティングエンティティを新しい活動ルーティングエンティティに変換するステップと、

(b) 活動切替えに続いて、新しい活動ルーティングエンティティと前記故障に関連するネットワークノードの各直接隣接する隣接ノードとの間で、トポロジ状態情報を交換し、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードのそれぞれが、同期化されたトポロジ状態情報を所有するようにするステップとを含む、

前記故障に関連するネットワークノードと各前記直接隣接する隣接ノードによって、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードとの間のトポロジ状態情報の交換が、前記故障に関連するネットワークノードと各前記直接隣接する隣接ノードにそれぞれ関係するときの局所的な状態情報の前記断続的な通告を中止せずに実施される方法。

【請求項2】 故障の前に、トポロジ状態情報を非活動ルーティングエンティティに送信するステップをさらに含み、それにより、活動ルーティングエンティティと非活動ルーティングエンティティの両方が、トポロジ状態情報の前記送信直後にネットワークのトポロジ状況全体の共通認識を共有する、請求項1に記載の方法。

【請求項3】 トポロジ状態情報の送信が周期的である、請求項2に記載の方法。

【請求項4】 非活動ルーティングエンティティへのトポロジ状態情報の周期的な送信が活動ルーティングエンティティからのものである、請求項3に記載の方法。

【請求項5】 故障の前に非活動ルーティングエンティティに局所的な状態情報を送信するステップをさらに含み、それにより、活動ルーティングエンティティと非活動ルーティングエンティティの両方が、局所的な状態情報の前記送信直後に局所的な状況の共通認識を共有する、請求項2に記載の方法。

【請求項6】 局所的な状態情報の送信が周期的であ

る、請求項5に記載の方法。

【請求項7】 非活動ルーティングエンティティへの局所的な状態情報の周期的な送信が活動ルーティングエンティティからのものである、請求項6に記載の方法。

【請求項8】 局所的な状態情報が、局所的なリンクの状況情報と局所的なノードの状況情報を備える、請求項5に記載の方法。

【請求項9】 局所的なリンクの状況情報が、リンクの特性、リンクの動作状況、およびポート識別子を備えるグループから選択される、請求項8に記載の方法。

【請求項10】 局所的なノードの状況情報が、ノード識別子、ピアグループ識別子、区分けされたノード選択状況、区分けされたノードリーダーシップ状況、および局所的な到達可能アドレスを備えるグループから選択される、請求項8に記載の方法。

【請求項11】 活動ルーティングエンティティと非活動ルーティングエンティティが、各々前記故障に関連するネットワークノードの一部を形成する、請求項7に記載の方法。

【請求項12】 活動ルーティングエンティティと非活動ルーティングエンティティが、各々別々の物理的構成要素によって実施される、請求項11に記載の方法。

【請求項13】 通信ネットワークが、非同期転送モード(ATM)ネットワークであり、通信ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルがPNNIプロトコルである、請求項5に記載の方法。

【請求項14】 通信ネットワークが、インターネットプロトコル(IP)ネットワークであり、通信ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルがOSPF(Open Shortest Path First: 空き最短パス第一)プロトコルである、請求項5に記載の方法。

【請求項15】 故障の前に、活動ルーティングエンティティから送信されたトポロジ状態情報が、活動ルーティングエンティティに関連付けられているトポロジデータベースから抽出され、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードの間の活動切替えに続いて交換されたトポロジ状態情報が、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードにそれぞれ関連付けられているトポロジデータベースから抽出される、請求項13に記載の方法。

【請求項16】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素(P T S E)にバンドルすることによって、故障の前に活動ルーティングエンティティから非活動ルーティングエンティティに送信される、請求項15に記載の方法。

【請求項17】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素(P T S E)にバンドルすることによって、新しい活動ルーティングエンティティ

10

20

30

40

50

ィと各前記直接隣接する隣接ノードの間の活動切替えに
続いて交換される、請求項15に記載の方法。

【請求項18】 各PTSEが、前記送信のために、P
NNIトポロジ状態パケット（PTSP）内にカプセル
化される、請求項16に記載の方法。

【請求項19】 各PTSEが、前記交換のために、P
NNIトポロジ状態パケット（PTSP）内にカプセル
化される、請求項17に記載の方法。

【請求項20】 新しい活動ルーティングエンティティ
が、各前記直接隣接する隣接ノードとのトポロジ状態情
報の前記交換の前に、前記交換が局所的な状態情報の前
記断続的な通告を前記中止せずに行われることを各前記
直接隣接する隣接ノードに通知する、請求項15に記載
の方法。

【請求項21】 局所的な状態情報の前記断続的な通告
の前記中止をせずにトポロジ状態情報の前記交換を通知
することが、前記故障に関連するネットワークノードの
各直接隣接する隣接ノードに送信された通知メッセージ
内のフラグによって行われる、請求項20に記載の方
法。

【請求項22】 通知メッセージが、前記フラグが提供
されたPNNIデータベースサマリパケットである、請
求項21に記載の方法。

【請求項23】 ネットワーク全体に局所的な状態情報
を断続的に通告するためのルーティングプロトコルを含
む、通信ネットワークにおける故障から回復するための
ネットワーク要素であって、

通信ネットワークに関するトポロジ状態情報に関連付
けられている活動ルーティングエンティティと、
非活動ルーティングエンティティであって、活動ルー
ティングエンティティの故障の際に、活動ルーティン
グエンティティと非活動ルーティングエンティティの間で活
動切替えが実行され、それにより、ネットワークの接続
を活動ルーティングエンティティから非活動ルーティ
ングエンティティに転換し、非活動ルーティングエン
ティティを新しい活動ルーティングエンティティに変換する
非活動ルーティングエンティティと、

データベース同期化プロセッサであって、活動切替えに
続いて、新しい活動ルーティングエンティティとネット
ワーク要素の各直接隣接する隣接ノードの間で、トポロ
ジ状態情報の交換を実施して、新しい活動ルーティン
グエンティティと各前記直接隣接する隣接ノードが、それ
ぞれ同期化されたトポロジ状態情報を所有するように
し、新しい活動ルーティングエンティティと各前記直接
隣接する隣接ノードの間のトポロジ状態情報の前記交換
が、前記故障に関連するネットワークノードと各前記直
接隣接する隣接ノードによって、ネットワーク要素と各
直接隣接する隣接ノードにそれぞれ関係しているときの
局所的な状態情報の前記断続的な通告を中止せずに実施
される、データベース同期化プロセッサとを備えるネッ

トワーク要素。

【請求項24】 トポロジ状態情報が、活動ルーティ
ングエンティティの故障の前に非活動ルーティングエン
ティティに送信され、そのため活動ルーティングエン
ティティと非活動ルーティングエンティティの両方が、ト
ポロジ状態情報の前記送信に続いてネットワークのトポ
ロジ状況全体の共通認識を共有する、請求項23に記載
のネットワーク要素。

【請求項25】 トポロジ状態情報の送信が周期的であ
る、請求項24に記載のネットワーク要素。

【請求項26】 非活動ルーティングエンティティへの
トポロジ状態情報の周期的な送信が活動ルーティング
エンティティからのものである、請求項25に記載のネッ
トワーク要素。

【請求項27】 局所的な状態情報が、活動ルーティ
ングエンティティの故障の前に非活動ルーティングエン
ティティに送信され、それにより、活動ルーティング
エンティティと非活動ルーティングエンティティの両方が、
局所的な状態情報の前記送信に直接続いて、局所的な状
況の共通認識を共有する、請求項24に記載のネット
ワーク要素。

【請求項28】 局所的な状態情報の送信が周期的であ
る、請求項27に記載のネットワーク要素。

【請求項29】 非活動ルーティングエンティティへの
局所的な状態情報の周期的な送信が活動ルーティング
エンティティからのものである、請求項28に記載のネッ
トワーク要素。

【請求項30】 局所的な状態情報が、リンクの状況情
報と局所的なノードの状況情報を備える、請求項24に
記載のネットワーク要素。

【請求項31】 局所的なリンクの状況情報が、リンク
の特性、リンクの動作状況、およびポート識別子を備え
るグループから選択される、請求項30に記載のネット
ワーク要素。

【請求項32】 局所的なノードの状況情報が、ノード
識別子、ピアグループ識別子、区分けされたノード選択
状況、区分けされたノードリーダーシップ状況、および
局所的な到達可能アドレスを備えるグループから選択さ
れる、請求項31に記載のネットワーク要素。

【請求項33】 活動ルーティグエンティティと非活動
ルーティングエンティティが、各々別々の物理構成要素
によって実施される、請求項32に記載のネットワーク
要素。

【請求項34】 通信ネットワークが、非同期転送モ
ード（ATM）ネットワークであり、通信ネットワークを
通じて局所的な状態情報を断続的に通告するためのルー
ティングプロトコルがPNNIプロトコルである、請求
項26に記載のネットワーク要素。

【請求項35】 通信ネットワークが、インターネット
プロトコル（IP）ネットワークであり、通信ネットワ

ークを通じてリンクの状況情報を断続的に通告するためのルーティングプロトコルがOSPF (Open Shortest Path First: 空き最短パス第一) プロトコルである、請求項26に記載のネットワーク要素。

【請求項36】 活動ルーティングエンティティの故障の前に、活動ルーティングエンティティから送信されたトポロジ状態情報が、活動ルーティングエンティティに関連付けられているトポロジデータベースから抽出され、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードの間の活動切替えに続いて交換されたトポロジ状態情報が、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードのそれぞれに関連付けられているトポロジデータベースから抽出された、請求項34に記載のネットワーク要素。

【請求項37】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素(PTSE)にバンドルすることによって、故障の前に活動ルーティングエンティティから非活動ルーティングエンティティに送信される、請求項23に記載のネットワーク要素。

【請求項38】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素(PTSE)にバンドルすることによって、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードの間の活動切替えに続いて交換される、請求項23に記載のネットワーク要素。

【請求項39】 各PTSEが、前記送信のためにPNNIトポロジ状態パケット(PTSP)内にカプセル化される、請求項37に記載のネットワーク要素。

【請求項40】 各PTSEが、前記交換のためにPNNIトポロジ状態パケット(PTSP)内にカプセル化される、請求項38に記載のネットワーク要素。

【請求項41】 各前記直接隣接する隣接ノードとのトポロジ状態情報の前記交換の前に、新しい活動ルーティングエンティティが、各前記直接隣接する隣接ノードに、前記交換が局所的な状態情報の前記断続的な通告の前記中止をせずに行われることを通知する、請求項23に記載のネットワーク要素。

【請求項42】 局所的な状態情報の前記断続的な通告の前記中止をせずにトポロジ状態情報の前記交換を通知することが、ネットワーク要素の各直接隣接する隣接ノードに送信された通知メッセージ内のフラグによって行われる、請求項41に記載のネットワーク要素。

【請求項43】 通知メッセージが、前記フラグが提供されたPNNIデータベースサマリパケットである、請求項42に記載のネットワーク要素。

【請求項44】 通信ネットワークにおける2つのネットワークノードの間でトポロジ状態情報を同期化するための方法であって、通信ネットワークが、ネットワークを通じて局所的な状態情報を断続的に通告するためのル

ーティングプロトコルを備え、2つのネットワークノードが、トポロジ状態を同期化する要求を開始する要求ノードと、前記要求を受信し、要求ノードと通信して、要求ノードが前記要求を開始したとき、要求ノードによって所有されていなかったトポロジ状態情報を要求ノードに提供する応答ノードとを備え、前記方法が、応答ノードに前記要求を行う前に、要求ノードによって、同期化の第1モードと第2モードの間で選択するステップを含み、第1前記モードが、前記要求ノードと前記応答ノードによって、要求ノードと応答ノードにそれぞれ関係しているときの局所的な情報の前記断続的な通告の中止を必要とするトポロジ状態の同期化を行ない、第2前記モードが、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を維持するトポロジ状態の同期化を行なう方法。

【請求項45】 トポロジ状態の同期化が、要求ノードによって応答ノードに対して要求された後に、要求ノードと応答ノードの間でトポロジ状態情報を交換するステップをさらに含む、請求項44に記載の方法。

【請求項46】 トポロジ状態情報の前記交換の前に、要求ノードが応答ノードに、前記交換が同期化の前記第1モードと前記第2モードの一方により行われることを通知する、請求項45に記載の方法。

【請求項47】 前記通知が、要求ノードによって応答ノードに送信された通知メッセージ内のフラグによって行われる、請求項46に記載の方法。

【請求項48】 要求ノードと応答ノードの間で交換されたトポロジ状態情報が、前記要求ノードと応答ノードにそれぞれ関連付けられているトポロジ状態データベースから抽出される、請求項47に記載の方法。

【請求項49】 通信ネットワークが非同期転送モード(ATM)ネットワークであり、ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルがPNNIプロトコルである、請求項47に記載の方法。

【請求項50】 通信ネットワークが、インターネットプロトコル(IP)ネットワークであり、ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルがOSPF (Open Shortest Path First: 空き最短パス第一) プロトコルである、請求項47に記載の方法。

【請求項51】 通知メッセージが、前記フラグが提供されたPNNIデータベースサマリパケットである、請求項49に記載の方法。

【請求項52】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素(PTSE)にバンドルすることによって交換される、請求項51に記載の方法。

【請求項53】 各PTSEがPNNIトポロジ状態パケット(PTSP)内にカプセル化される、請求項52

に記載の方法。

【請求項54】 通信ネットワークにおける2つのネットワークノードの間でトポロジ状態情報を同期化するためのネットワーク要素であって、通信ネットワークが、ネットワークを通じて、局所的な状態情報を断続的に通告するためのルーティングプロトコルを備え、2つのネットワークノードが、トポロジ状態を同期化する要求を開始する要求ノードと、前記要求を受信し、要求ノードと通信して、要求ノードが前記要求を開始したとき、要求ノードによって所有されていなかったトポロジ状態情報を要求ノードに提供する応答ノードとを備え、ネットワーク要素が同期化の2つのモードの一方において選択的に動作し、第1モードが、要求ノードと応答ノードの間で、前記要求ノードと前記応答ノードによって、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告の中止を必要とするトポロジ状態の同期化を実施し、第2モードが、要求ノードと応答ノードの間で、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を維持するトポロジ状態の同期化を実施するネットワーク要素。

【請求項55】 トポロジ状態の同期化が要求ノードによって応答ノードに対して要求された後に、トポロジ状態情報が要求ノードと応答ノードの間で交換される、請求項54に記載のネットワーク要素。

【請求項56】 トポロジ状態情報の前記交換の前に、要求ノードが応答ノードに、前記交換が同期化の前記第1モードと前記第2モードの一方により行われることを通知する、請求項55に記載のネットワーク要素。

【請求項57】 前記通知が、要求ノードによって応答ノードに送信された通知メッセージ内のフラグによって行われる、請求項56に記載のネットワーク要素。

【請求項58】 要求ノードと応答ノードの間で交換されたトポロジ状態情報が、各々前記要求および応答ノードにそれぞれ関連付けられているトポロジ状態データベースから抽出される、請求項56に記載のネットワーク要素。

【請求項59】 通信ネットワークが、非同期転送モード(ATM)ネットワークであり、ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルがPNNIプロトコルである、請求項57に記載のネットワーク要素。

【請求項60】 通信ネットワークがインターネットプロトコル(IP)ネットワークであり、ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルがOSPF(Open Shortest Path First)プロトコルである、請求項57に記載のネットワーク要素。

【請求項61】 通知メッセージが、前記フラグが準備されたPNNIデータベースサマリパケットである、請

求項59に記載のネットワーク要素。

【請求項62】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素(PTSE)にバンドルすることによって交換される、請求項61に記載のネットワーク要素。

【請求項63】 各PTSEがPNNIトポロジ状態パケット(PTSP)内にカプセル化される、請求項62に記載のネットワーク要素。

【請求項64】 通信ネットワークの第1ネットワークノードと第2ネットワークノードの間でトポロジ状態情報を同期化するための方法であって、通信ネットワークが、ネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルを備え、第1ネットワークノードが、トポロジ状態を同期化する要求を開始し、第2ネットワークノードが、前記要求を受信し、第1ネットワークノードと通信して、第1ネットワークノードにトポロジ状態情報を提供し、トポロジ状態の同期化が第1モードにより行われ、第1ネットワークノードと第2ネットワークノードにそれぞれ関係しているときの局所的な状態情報の前記交換が中止されない方法。

【請求項65】 トポロジ状態の同期化の第2モードが提供され、トポロジ状態の同期化の第2モードが、第1ネットワークノードと第2ネットワークノードによって、第1ネットワークノードと第2ネットワークノードにそれぞれ関係しているときの局所的な状態情報の前記交換を中止することを必要とする、請求項64に記載の方法。

【請求項66】 さらに、第1ネットワークノードによって第2ネットワークノードに前記要求が開始される前に、トポロジ状態の同期化の第1モードとトポロジ状態の同期化の第2モードとの間で選択するステップを含む、請求項65に記載の方法。

【請求項67】 第2ネットワークノードが第1ネットワークノードと通信して、第1ネットワークノードが前記要求を開始したときに第1ネットワークノードによって所有されていなかったトポロジ状態情報を第1ネットワークノードに提供する、請求項66に記載の方法。

【請求項68】 トポロジ状態情報の前記交換の前に、第1ネットワークノードが第2ネットワークノードに、トポロジ状態情報の前記交換が同期化の前記第1モードと前記第2モードの一方により行われることを通知する、請求項67に記載の方法。

【請求項69】 前記通知が、第1ネットワークノードによって第2ネットワークノードに送信された通知メッセージのフラグによって行われる、請求項68に記載の方法。

【請求項70】 第1ネットワークノードと第2ネットワークノードの間で交換されるトポロジ状態情報が、各々前記第1および第2ネットワークノードにそれぞれ関連付けられているトポロジ状態データベースから抽出さ

10

20

30

40

50

れる、請求項69に記載の方法。

【請求項71】 通信ネットワークが非同期転送モード（ATM）ネットワークであり、ネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルがPNNIプロトコルである、請求項70に記載の方法。

【請求項72】 通信ネットワークがインターネットプロトコル（IP）ネットワークであり、ネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルがOSPF（Open Shortest Path First：空き最短パス第一）プロトコルである、請求項70に記載の方法。

【請求項73】 通知メッセージが、前記フラグが提供されたPNNIデータベースサマリパケットである、請求項71に記載の方法。

【請求項74】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素（PTSE）にバンドルすることによって交換される、請求項73に記載の方法。

【請求項75】 各PTSEがPNNIトポロジ状態パケット（PTSP）内にカプセル化される、請求項74に記載の方法。

【請求項76】 通信ネットワークの第1ネットワークノードと第2ネットワークノードの間でトポロジ状態情報を同期化するためのネットワーク要素であって、通信ネットワークが、ネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルを備え、第1ネットワークノードがトポロジ状態を同期化する要求を開始し、第2ネットワークノードが前記要求を受信し、第1ネットワークノードと通信して、トポロジ状態情報を第1ネットワークノードに提供し、トポロジ状態の同期化が第1モードにより行われ、第1ネットワークノードと第2ネットワークノードにそれぞれ関係しているときの局所的な状態情報の前記交換が中止されない、ネットワーク要素。

【請求項77】 トポロジ状態の同期化の第2モードが提供され、トポロジ状態の同期化の第2モードが、第1ネットワークノードと第2ネットワークノードによって、第1ネットワークノードと第2ネットワークノードにそれぞれ関係しているときの局所的な状態情報の前記交換を中止することを必要とし、ネットワーク要素が、トポロジ状態の同期化の2つの前記モードの一方において選択的に動作する、請求項76に記載のネットワーク要素。

【請求項78】 第1ネットワークノードによって第2ネットワークノードに前記要求を開始する前に、ネットワーク要素がトポロジ状態の同期化の第1モードとトポロジ状態の同期化の第2モードの間で選択する、請求項77に記載のネットワーク要素。

【請求項79】 第1ネットワークノードによって第2

ネットワークノードに前記要求を開始する前に、第1ネットワークノードが第2ネットワークノードに、トポロジ状態情報の前記交換がトポロジ状態の同期化の第1モードとトポロジ状態の同期化の第2モードの一方により行われることを通知する、請求項78に記載のネットワーク要素。

【請求項80】 第2ネットワークノードが第1ネットワークノードと通信して、第1ネットワークノードが前記要求を開始したときに、第1ネットワークノードによって所有されていなかったトポロジ状態情報を第1ネットワークノードに提供する、請求項79に記載のネットワーク要素。

【請求項81】 前記通知が、第1ネットワークノードによって第2ネットワークノードに送信された通知メッセージ内のフラグによって行われる、請求項80に記載の方法。

【請求項82】 第1ネットワークノードと第2ネットワークノードの間で交換されるトポロジ状態情報が、各々前記第1および第2ネットワークノードにそれぞれ関連付けられているトポロジ状態データベースから抽出される、請求項81に記載の方法。

【請求項83】 通信ネットワークが非同期転送モード（ATM）ネットワークであり、ネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルがPNNIプロトコルである、請求項82に記載の方法。

【請求項84】 通信ネットワークがインターネットプロトコル（IP）ネットワークであり、ネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルがOSPF（Open Shortest Path First：空き最短パス第一）プロトコルである、請求項82に記載の方法。

【請求項85】 通知メッセージが、前記フラグが提供されたPNNIデータベースサマリパケットである、請求項83に記載の方法。

【請求項86】 トポロジ状態情報が、トポロジ状態情報をPNNIトポロジ状態要素（PTSE）にバンドルすることによって交換される、請求項85に記載の方法。

【請求項87】 各PTSEがPNNIトポロジ状態パケット（PTSP）内にカプセル化される、請求項86に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、一般に、トポロジ状態ルーティングプロトコルを有する通信ネットワークでのネットワークトポロジデータベースの再同期化の分野に関し、より具体的には、そのようなネットワークでネットワークトポロジデータベースの再同期化を達成するための方法および装置に関する。例えば、本発明は、

ネットワークのノードに関連する活動ルーティングエンティティに影響を与えるノードの故障またはリセットに続く冗長性回復の状況におけるデータベースの再同期化に非常に適している。

【0002】

【従来の技術】トポロジ状態ルーティングプロトコルは通信ネットワークで使用され、そのようなネットワーク内のノード間およびノードクラス間にトポロジ状態の情報を伝達または通告する。通告されたトポロジ状態の情報は、順次所与のネットワーク全体の通信の最適な経路を計算するために使用される。本出願で使用されるように、トポロジ状態情報を参照することにより、ネットワークドメイン全体としての状態情報が示される。あるネットワークプロトコルでは、トポロジ状態情報はリンクの状態情報とノードの状態情報の両方を含む。例えば、リンクの状態情報は、リンクの特性、リンクの動作状況、ポート識別子、および、隣接する隣接ノードに関する遠隔隣接ノード情報などの属性を含む。ノードの状態情報は、ノード識別子、ピアグループ識別子、分けされたノードの選択状況、分けされたノードのリーダーシップ状況、および局所的到達可能なアドレス情報などの属性を含む。

【0003】トポロジ状態情報はネットワークドメイン全体としての状態情報を指すが、本出願では、特定のネットワークノードによって局所的に発信される状態情報を取り扱う際に、局所的な状態情報を参照する。局所的なリンク状況情報は、ピアノードとの通信状況に関する所与のノードの認識を反映することになる。したがって、局所的なリンク状況情報も、トポロジリンク状況情報と同様に、リンクの特性、リンクの動作状況、ポート識別子、および隣接する隣接ノードに関する遠隔隣接情報などの属性を含むが、これらは、ネットワークドメインの一部を形成する様々なノードとは異なり、所与のネットワークノードに関係することになる。同様に、局所的なノードの状態情報は、ノード識別子、ピアグループ識別子、分けされたノードの選択状況、特別なノードのリーダーシップ状況、および局所的な到達可能なアドレス情報などの属性を含む。ここでも、これらは、トポロジのノード状態情報を参照する際に、ネットワークドメイン全体に関係する代わりに、局所的なノードの状態を参照する際に、所与のノードに関係することになる。本出願では、状態情報を参照することにより、トポロジ状態情報と局所的な状態情報の両方が示される。

【0004】いくつかの知られているトポロジ状態プロトコルでは、通信ネットワークにある特定のノードは、そのネットワークのルーティング機能を適切に動作させるために、特別なまたは追加の責務を担う可能性がある。例えば、1998年4月のJ. Moyによる「OSPF Version 2」、STD 54、RFC 2328に記述されているオープンショーテストパスファース

ト(OSPF) IPルーティングプロトコルでは、指定ルータ(DR)として認定されたノードが、そのような責務を帯びることになる。同様に、プライベートネットワークノードインタフェースまたはプライベートネットワークトウネットワークインタフェース(PNNI)プロトコルでは、ピアグループリーダー(PGL)という名称のノードが、この性質の責務を帯びている。PNNIプロトコルは、(i) 1996年3月の「Private Network Interface Specification Version 1.0: 専用ネットワークインタフェース仕様書1.0版」、ATM Forum document no. af-pnni-0055.000、(ii) 1996年9月の「Private Network-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB) 専用ネットワークネットワークインタフェース仕様書1.0版 補遺(Soft PVC MIB)」、ATM Forum document no. af-pnni-0066.000、(iii) 1997年1月の「Addendum to PNNI V1.0 for ABR parameter negotiation」、ATM Forum document no. af-pnni-0075.000という名称の文献、ならびに(iv) 1997年5月の「PNNI V1.0 Errata and PICS、ATM Forum document no. af-pnni-0081.000に見られる修正において指定されている(これ以降、以上の文献(i)から(iv)のすべては、包括的に一括して「PNNI仕様」と呼ぶ)。PNNI仕様は、参照により本明細書に組み込まれる。

【0005】ネットワーク空間内にある所与の物理ノードは、分散選択として知られているプロセスによって、上述したタイプの特別なネットワークの責務を獲得することが可能である。分散選択のスキームでは、ネットワーク階層の特定のレベルにあるすべてのノードは、トポロジ状態プロトコルに関して追加のタスクまたは責務を担うべきノードを選択するために通信する。当業者なら、分散選択のプロセスを実施することは、特定のネットワーク環境に応じてかかる時間が変わることを理解するであろう。そのうえ、ダウンタイムのために、この特別な立場が所与のネットワークノードによって埋められない場合、ネットワークの一部またはネットワークドメイン全体のルーティング機能は、ダウンタイムの合間に能力の低下または非効率を呈す可能性がある。したがって、トポロジ状態プロトコルを使用する通信ネットワークでは、ネットワークノードの故障の後にネットワークルーティングシステムが回復時間間隔を許容しなければならない。例えばこれは、重大性の程度は変わるが、故

障したノードが先に言及した追加の責務を有する選択されたネットワークノードの機能に影響を与えれば生じる可能性がある。

【0006】あるルーティングプロトコルは、所与のレベルのノード冗長性を指定する。この冗長性は、上述した種類の特別のプロトコル機能を実施するノードに影響を与える故障の場合に、ネットワークルーティングシステムの回復時間を低減することを意図している。例えば、OSPFプロトコルでは、バックアップ指定ルータ(BDR)の使用が指定されている。バックアップ指定ルータに命令して、現在指定されている指定ルータに影響を与える故障を検出させる。そのような故障を検出した時点で、バックアップ指定ルータは回復アクションを取ることを要求され、故障した以前の指定ルータの代わりに、自身を新しい指定ルータに宣言する。共有ネットワークの影響を受けた部分上にあるすべての他のルータは、その後、新しい指定ルータノードの存在を通知される。したがって、指定ルータノードに影響を与えた故障に続いて、OSPFプロトコルの下で動的選択プロセスを再実行する必要はないが、それにも関わらず、ある期間のネットワークルーティングの停止は、当初故障した指定ルータノードによりサービスされていた共有ネットワーク上のすべてのルータとホストが経験することになる。これは、影響を受けたルータとホストが、関連する指定ルータノードに影響を与える故障に続いて、ネットワークルーティングシステムの回復機能に携わっているからである。

【0007】一方、PNNIプロトコルでは、特別ノードの冗長性に対して、現在準備は行われていない。したがって、分散選択プロセスとそれに関連するプロトコル動作は、特別のネットワークノードに影響を与える故障が発生した時点で再実行されなければならない。PNNIプロトコルでは、トポロジ階層の1つのレベルにおいてピアグループリーダー機能を実施する物理ノードは、この機能を階層のいくつかの他のレベルで実施していることがある。したがって、そのような物理ノードに影響を与える故障は、集合ネットワークの大部分に非常に影響を与える可能性がある。さらに、現在のPNNIプロトコルでは、バックアップピアグループリーダーが提供されていない。したがって、上述した種類の複数レベルのピアグループリーダーに影響を与える故障は、複数レベルのピアグループリーダーによって代表される様々なピアグループの一部を形成するすべての論理ノードによって検出されなければならない。ネットワーク階層の異なるレベルにあるこれらの論理ノードは、その後、新しいピアグループリーダーを選択しなければならない。OSPFプロトコルに関連して上記で示した例の場合のように、ピアグループリーダーの故障は、多くのノードが知ることができるので、そのようなノードは、概ねすべて、ルーティングシステムの影響を受けた機能を回復す

ることに携わらなければならない。これを考慮すると、PNNIネットワークにおけるピアグループリーダーの故障は、ネットワークの大きな部分に影響を与え、多くの場合、一定の間、サービスプロバイダまたはエンドユーザにとって許容し得ないネットワークのルーティング挙動の混乱を生じる可能性があることが考えられる。

【0008】上記の議論は、特別の責務を有するネットワークノードに影響を与える故障の影響を明らかにした。しかし、当業者なら、特別の責務を有していない通常の物理ノードまたは論理ノードに関する故障も、故障した通常のノードによってサービスされていた隣接ノードまたはデバイスのルーティング機能にある程度の混乱をもたらすことになることが理解されよう。あるノードのアーテクチャでは、ルーティング機能が故障した場合に、パケット転送またはコール処理などのあるネットワーク機能を保持することが可能であろうが、OSPFおよびPNNIなどのトポロジ状態プロトコルは、ドメインの各ネットワークノードが、ルーティングシステムに参加する前にトポロジデータベースをその近隣と同期化することを要求する。そのようなトポロジデータベースの同期化は、ノードの故障から回復するために、これらのネットワークプロトコルで行われなければならない。同期化のプロセスは、状況に応じて、回復のスキーム全体で、数秒から数分かかる可能性がある。同期化中、故障したノードによってサービスされていたネットワークのデバイスは影響を受けることになり、したがって、ルーティング機能は、非常に混乱する可能性がある。上記の議論は、ノードの故障から回復することにまつわる課題を主眼としているが、当業者なら、ネットワークノードに関連付けられているルーティングプロセッサのリセットなど、ノードがトポロジデータベースの同期化を行うことを要求する他のイベントから類似の問題が生じることを理解できよう。

【0009】故障したルータを使用するホストにとってトランスペアレントな方式で、別々のルータ間の切替えを保証するある機構が従来の技術で開発されている。1998年3月のT. Li, B. Cole, P. Morton、およびD. Liによる「Cisco Hot Standby Router Protocol (HSRP)」RFC 2281、に記述されているホットスタンバイルータプロトコル、および1997年冬のP. HigginsonおよびM. Shandの「Development of Router Clusters to Provide Fast Failover in IP Networks: IPネットワークにおける高速障害回復を実行するルータラスタの開発」、9 Digital Technical Journal、No. 3によるIPスタンバイプロトコルは、そのようなトランスペアレントなルータ切替えスキームの2つの例である。しかし、以下でより詳細に説明するよ

うに、この種の切替え機構は、一般に、故障したノードに直接隣接する特定のホストまたはノードを超えて、切替えがネットワークのルータまたはノードに広くトランスペアレントであることを保証しない。従来の技術では、ノードの故障は通常、別々の異なるノードによって回復される。したがって、ノードのルーティング構成要素の故障を、直近の隣接ノード以外のすべてのノードにトランスペアレントな方式で、同じノードの他のルーティング構成要素によって回復することを可能にする機構を提供することが有利である。

【0010】したがって、従来の技術のトポロジ状態ルーティングプロトコルは、ノードの故障から回復する状況、または、ノードが、以前にそのトポロジデータベースを同期化した後、またそれを行うことを要求する可能性がある他の状況に直面すると問題および課題を生じ、これらの問題および課題は、故障によって即座に影響をされたノードが特別の責務を有するか否かに関わらず発生する。第1に、知られている回復機構は、通常、ネットワークの少なくとも一部のルーティング機能を混乱させ、ネットワークを使用するあるデバイスにサービス衝撃を与える。影響を受けたネットワークの部分は、環境により異なることになる。例えば、ネットワークの影響を受けた部分は、特別の機能を実施するノードのほうかそのような機能を実施しないノードの場合より広範囲にわたることが予期できる。そのうえ、影響を受けた部分は、OSPF指定ルータに影響を与える故障より、PNNIPピアグループリーダーに関する故障のほうがより広大であることが予期できる。第2に、ノードまたはリンクの故障から回復するのに必要な時間は変動するが、最高で数分またはそれ以上の程度である可能性がある。上述したように、この時間枠は、あるサービスプロバイダまたはエンドユーザが許容できない可能性がある。第3に、多くのノードが故障に気づかなければならず、したがって、回復プロセスに携わることが必要なので、帯域幅と処理時間の性質をおびたネットワークリソースが転用されることになる。これは、一般に他のネットワークの活動を損ね、特にネットワークルーティングシステムの性能と安定性を低減する可能性がある。

【0011】

【発明が解決しようとする課題】したがって、本発明の目的は、一般に、トポロジ状態ルーティングプロトコルを有するネットワークでデータベースを再同期化するための方法と装置、具体的にはネットワークノードのルーティングエンティティに関連付けられているノードの故障に続いて、特に冗長性回復の状況に適している方法と装置を提供することを追求することであり、それに準じて、代替的な従来の技術およびデバイスによって提示された問題のいくつかを、いくつかの事例については、軽減または克服することが可能である。

【0012】

【課題を解決するための手段】本発明の第1の広範な態様によれば、通信ネットワークの活動ルーティングエンティティに影響を与える故障から回復するための方法が提供されている。活動ルーティングエンティティは、通信ネットワークのネットワークノードに関連付けられており、通信ネットワークは、ネットワークを通じて、局所的な状態情報を断続的に通告するためのルーティングプロトコルを備え、さらに、故障した際に、ネットワークノードのネットワーク接続を、活動ルーティングエンティティから転換することができる非活動ルーティングエンティティを備え、その方法は、(a)故障した際に、活動ルーティングエンティティと非活動ルーティングエンティティの間で活動切替えを実行し、ネットワークノードのネットワーク接続が、活動ルーティングエンティティから非活動ルーティングエンティティに転換され、それにより、非活動ルーティングエンティティを新しい活動ルーティングエンティティに変換するステップと、(b)活動切替えに続いて、新しい活動ルーティングエンティティと前記故障に関連するネットワークノードの各直接隣接する隣接ノードとの間でトポロジ状態情報を交換し、したがって、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードが、それぞれ、同期化したトポロジ状態情報を所有するようにし、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードとの間のトポロジ状態情報の交換が、前記故障に関連するネットワークノードと各前記直接隣接する隣接ノードとによって、前記故障に関連するネットワークノードと各前記直接隣接する隣接ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を中止せずに実施されるステップとを含む。

【0013】本発明の第2の広範な態様によれば、ネットワークを通じて、局所的な状態情報を断続的に通告するためのルーティングプロトコルを含む通信ネットワークにおける故障から回復するためのネットワーク要素が提供されている。ネットワーク要素は、通信ネットワークに関するトポロジ状態情報に関連付けられている活動ルーティングエンティティと、活動ルーティングエンティティの故障した際に、活動切替えが活動ルーティングエンティティと非活動ルーティングエンティティの間で実行され、それにより、ネットワーク接続を活動ルーティングエンティティから非活動ルーティングエンティティに転換し、非活動ルーティングエンティティを新しい活動ルーティングエンティティに変換する非活動ルーティングエンティティと、活動切替えに続いて新しい活動ルーティングエンティティとネットワーク要素の各直接隣接する隣接ノードとの間でトポロジ状態情報の交換を実施して、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードが、それぞれ同期化したトポロジ状態情報を所有するようにし、新しい活動ルーティングエンティティと各前記直接隣接する隣接ノードと

の間のトポロジ状態情報の前記交換が、前記故障に関連するネットワークノードと各前記直接隣接する隣接ノードとによって、ネットワーク要素と各直接隣接する隣接ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を中止せずに実施されるデータベース同期化プロセッサとを備える。

【0014】本発明の第3の広範な態様によれば、通信ネットワークにおける2つのネットワークノードの間でトポロジ状態情報を同期化するための方法が提供されている。通信ネットワークは、ネットワークを通じて局所的な状態情報を断続的に通告するためのルーティングプロトコルを備え、2つのネットワークノードは、トポロジ状態を同期化する要求を開始する要求ノードと、前記要求を受信し、要求ノードと通信して、要求ノードが前記要求を開始したときに要求ノードによって所有されていないトポロジ状態情報を要求ノードに提供する応答ノードとを備え、その方法は、前記要求が応答ノードに対して要求ノードによって作成される前に、同期化の第1モードと第2モードの間で選択するステップを備え、第1前記モードは、前記要求ノードと前記応答ノードによって、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を中止することを必要とするトポロジ状態の同期化を行ない、第2前記モードは、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を維持するトポロジ状態の同期化を行なう。

【0015】本発明の第4の広範な態様によれば、通信ネットワークにおける2つのネットワークノードの間でトポロジ状態情報を同期化するためのネットワーク要素が提供されている。通信ネットワークは、ネットワーク全体で局所的な状態情報を断続的に通告するためのルーティングプロトコルを備え、2つのネットワークノードは、トポロジ状態を同期化する要求を開始する要求ノードと、前記要求を受信し、要求ノードと通信して、要求ノードが前記要求を開始したときに、要求ノードによって所有されていないトポロジ状態情報を要求ノードに提供する応答ノードとを備え、ネットワーク要素は、同期化の2つのモードの一方で選択的に動作し、第1モードは、要求ノードと応答ノードの間で、前記要求ノードと前記応答ノードによって、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な中止を必要とするトポロジ状態の同期化を実施し、第2モードは、要求ノードと応答ノードの間で、要求ノードと応答ノードにそれぞれ関係しているときの局所的な状態情報の前記断続的な通告を維持するトポロジ状態の同期化を実施する。

【0016】本発明の第5の広範な態様によれば、通信ネットワークにおける第1ネットワークノードと第2ネットワークノードの間で、トポロジ状態情報の同期化を実施する方法が提供されている。通信ネットワークはネ

ットワーク全体で局所的な状態情報を交換するためのルーティングプロトコルを備え、第1ネットワークノードはトポロジ状態を同期化する要求を開始し、第2ネットワークノードは前記要求を受信し、第1ネットワークノードと通信して、トポロジ状態情報を第1ネットワークノードに提供し、トポロジ状態の同期化は、第1モードにより行われ、第1ネットワークノードと第2ネットワークノードにそれぞれ関係しているときの局所的な状態情報の前記交換は、中止されない。

10 【0017】本発明の第6の広範な態様によれば、通信ネットワークにおける第1ネットワークノードと第2ネットワークノードの間で、トポロジ状態情報を同期化するためのネットワーク要素が提供されている。通信ネットワークはネットワークを通じて局所的な状態情報を交換するためのルーティングプロトコルを備え、第1ネットワークノードはトポロジ状態を同期化する要求を開始し、第2ネットワークノードは前記要求を受信し、第1ネットワークノードと通信して、トポロジ状態情報を第1ネットワークノードに提供し、トポロジ状態の同期化が第1モードにより行われ、第1ネットワークノードと第2ネットワークノードにそれぞれ関係しているときの前記局所的な状態情報の交換は中止されない。

【0018】

【発明の実施の形態】一般に、ホット冗長性技術など、ネットワークの構成要素またはデバイスのための冗長性技術は、当業者によく知られている。図1を参照すると、これらの技術が、PNNIネットワークドメイン30の形態にある通信ネットワークの例示的な例を使用して説明されている。しかし、当業者なら、本発明は、局所的な状態情報の断続的な通告がOSPF (Open Shortest Path First: 空き最短パス第一) ルーティングプロトコルによって実施されるインターネットプロトコル(IP)ネットワークなど、他のタイプのネットワークにも同様に適用または適応することが可能であることを理解するであろう。そのうえ、本発明は、ネットワークノードのルーティングエンティティに関連する故障からの回復の状況だけでなく、ネットワークノードがそのトポロジデータベースを再同期化することが必要または望ましい可能性がある他の状況にも適している。

40 【0019】トポロジ状態ルーティングプロトコルおよびトポロジデータベースの同期化通信ネットワーク2は、各々通常交換機である、複数のネットワークノード32から41を備える、ネットワークドメイン30を有する。ネットワークノード32から41は、ネットワークドメインの2つの所与の交換機をそれぞれ接続する、物理リンクまたは論理リンク42から53によって、相互接続されている。PNNIネットワークドメイン30のネットワーク要素またはノード56 (または名称「A. 1. 2」) は、「PG (A)」という名称である

親ピアグループのピアグループリーダーの役割りを担っていることが示されており、親ピアグループのレベルにあるノード36の存在は、ノード56のリーダー資格の結果である。また、ノード36は、より低いレベルのネットワークノード56から60を備える子ピアグループ55（または名称「PG(A.1)」）の形態にあるネットワークドメインを表す。より低いレベルのネットワークノード56から60は、各々2つの所与のより低いレベルの交換機を接続する物理リンクまたは論理リンク62から67によって相互接続されている。PG(A.1)のピアグループリーダーを定義する機能は、より低いレベルのノード56（また名称「A.1.2」）を含む交換機の上で実施される。PG(A.1)は、PG(A)の子ピアグループであり、論理ノード36としてPG(A)に示されており、物理的な交換機56内で実施される。同様に、「PG(A)」という名称の親ピアグループは、それ自体は、単一論理ノード（図示せず）によってルーティング階層のより高いレベルで示されている子ピアグループとすることが可能である。

【0020】知られている冗長性技術によれば、耐故障性保護が望ましい特定のノード、スイッチ、または他のネットワークエンティティは、通常、単一ネットワーク要素内にある少なくとも2つのルーティングプロセッサを提供する。ルーティングプロセッサは、隣接する隣接ノードへの接続性を維持し、かつそれらのノードとトポロジ状態情報を共有する機能を実行する。ルーティングプロセッサは、別々の物理構成要素によって構成されることが好ましい。例えば、物理構成要素は、各々、ネットワークノード56（「A.1.2」）内など、同じネットワークスイッチ内に用意された別々のハードウェアカードの形態にあることが可能である。2つのプロセッサが冗長性のために提供されている場合、当該の物理構成要素の一方は、冗長ネットワーク要素のための活動ルーティングエンティティの役割りを担い、したがって、物理構成要素の他方は、冗長ネットワーク要素のための非活動ルーティングエンティティの役割りを担う。

【0021】活動ルーティングエンティティの故障を検出する際に、非活動ルーティングエンティティがサービスに呼び出され、故障した活動ルーティングエンティティの機能を引き継ぐ。この手順は、活動切替えと呼ばれる。これらのルーティングエンティティは、両方とも、同じノード（ネットワークノード56など）に関連付けられているので、ノード自体は、何ら特別の責務を放棄する必要はない。同様に、直接隣接する親ピアグループノード（ネットワークノード34、35、37、38）とあらゆる直接隣接する子ピアグループノード（ネットワークノード57、59、60）の形態にある故障したノードの直近の隣接ノードのみを呼び出す必要があるか、またはネットワークの回復に携わるようにリスト化する必要がある。しかし、以下で議論するように、現在

のトポロジ状態プロトコルは、それにも関わらず、回復プロセス中に、故障したノードの直近の隣接のノードより多くのノード（例えば、ネットワークノード32、33、39、40、41、58）に影響を与え、それにより、回復を行うのに必要な時間ならびにプロセスにおいて消費されるネットワークのリソースが増大する可能性がある。

【0022】既存の機能と技術を使用して、PNNIネットワークドメイン30など、所与のネットワークアーキテクチャにおいて、冗長保護のスキームを実施することが可能である。例えば、これらの機能と技術は、様々なネットワークノード内の活動状況の管理と、活動ルーティング構成要素と非活動ルーティング構成要素の間における状態情報の同期化を含むことが可能である。ネットワークトポロジのこの状態情報は、通常、ルーティングドメインの各ネットワークノードと関連付けられ、トポロジデータベースとも呼ばれる同期化データベースに格納されている。通常、同期化データベースは、当該のネットワークノード内に格納されることになる。データベースの同期化は、ネットワーク内の隣接ノードが、ネットワークのトポロジ全体の共通な見解を共有することを保証する既存のトポロジ状態ルーティングプロトコル機構である。ITU-TQ.2931など、いくつかのシグナリングプロトコルは、2つのネットワークノードの間で呼状態の同期化を実施する状況照会スキームなどの機構を有する。

【0023】冗長性のいくつかの知られているスキームに関する1つの問題は、故障が、より高いレベルのノード36を実施するノード56などのネットワークノードにおいて発生したとき、故障したノードに対して影響を与えおよびそれから影響を受けたPG(A)44、45、47、51のリンクが、ある時間の後、またはPG(A.1)の新しいPGLが、より高いレベルのノード36を実施する責務の引き継ぎを開始する間、通告を受けるのを停止することである。すなわち、新しい活動ルーティングプロセッサが、その仲間とデータベースの同期化を開始するとき、現在のPNNIプロトコルは、同期化に含まれている各ノードから局所的な状態情報を通告することを、同期化が行われる時間まで、除去または中止することを要求する。したがって、故障したノードは局所的な状態情報の通告を停止し、故障したノードの隣接ノードも、同様にそれぞれの局所的な状態情報の通告を停止することになる。これは、現在知られているプロトコルの下で実施されるあらゆる同期化に当てはまり、活動切替えまたはプロセッサのリセット状態の後、同期化が必要とされる。既存のPNNIプロトコルでは、ホット冗長性機能を提供する故障したノードは、そのトポロジデータベースを隣接ノードと同期化することが必要とされる。これは、そのようなノードの活動ルーティングプロセッサと非活動ルーティングプロセッサ

が、一般に、故障の前に内部でトポロジ状態情報を交換しているからであり、この交換は周期的に行われる。したがって、状態情報を最後に交換してから故障までの間にトポロジ状況情報が損失される可能性が非常にあり、この情報の損失は、以前は活動ルーティングプロセッサであり故障したルーティングプロセッサから活動切替えをした際に、非活動ルーティングプロセスが被る。

【0024】やはりより高いレベルのノード36を実施し、従来の技術による冗長機能を備える、ノード56などの故障したノードが、活動切替えによって再始動するとき、活動切替えにより得られる新しい活動ルーティングエンティティは、したがって、影響を受けたノード56の出入りリンク62、63、64、および影響を受けたより高いレベルのノード36の出入りリンク44、45、47、51を再確立しなければならない。ノード56などの故障したノードが、所与の数の子ピアグループノード56、57、58、59、および60に対するピアグループリーダーである場合、子ピアグループの他のリンクも再確立されなければならない。ネットワークトポロジの一部としてそれらの存在を再確立するために、故障したノードとその隣接ノードはまず、各々他の存在を認識しなければならない。PNNIプロトコルでは、物理リンクまたはVPCによって接続されている最低レベルのピアの場合に、両方向のハローパケットがこの隣接ノード発見の機能を達成する。次に、故障したノードとその各隣接ノードは、互いにトポロジデータベースサマリ情報を交換する。通常PNNIプロトコルでは、新しく知り合ったノードがまず、PNNIトポロジ状態要素ヘッダ(PTSE)を交換して、ノードが同期化されているか否かを決定する。同期化されていない場合、同期化が行われる。ノードが、まだ有していないPTSEを通告するPTSEヘッダ情報を受信するとき、通告されたPTSEを要求して、それを受信した後、要求したPTSEでトポロジデータベースを更新する。

【0025】同期化が完了した後、局所的な状態情報は、ネットワークドメイン30のネットワークトポロジ内の故障したノードとその隣接ノードの間で通告され、それにより、各ノードは、すべての他の隣接ノードへの到達可能性を通告する。例えば、PNNIプロトコルでは、この通告は、フラッディング(flooding)として知られている通常のプロセスによって行われる。以前に説明したように、この再確立の知られているプロセスは、ネットワークのルーティング機能に悪い影響を与える。さらに、ネットワークのリソースは、知られている技術により、故障したノード56の再確立中に消費される。

【0026】知られている隣接ピア有限状態機械、隣接ピアデータ構造、およびデータサマリパケット構造に対する変更

例示的な実施形態によれば、本発明は、図2の既存のプ

ロトコルに様々な変更を行うことによって、PNNIプロトコルの状況で採用することが可能である。以下でより詳細に説明するように、これらの変更は、一般に、知られている隣接ピアの有限状態機械(FSM)5とそれに関連する遷移イベント、隣接ピアデータ構造の他の状態、およびデータベースサマリパケット構造に関係する。図3を参照すると、データベースの再同期化を実施するための2つの追加の状態が、変更した隣接ピア有限状態機械10に対して定義されている。追加の状態は、フル状態での交換(Exchange in Full State)22およびフル状態でのローディング>Loading in Full State)24と名付けられている。フル状態での交換(Exchange in Full State)22、フル状態でのローディング>Loading in Full State)24、およびそれらの関連する状態遷移について、以下でより詳細に説明する。

【0027】隣接ピア状態の変化を生じさせる新しい遷移イベントも、PNNI仕様のセクション5.7.3に見られる既存の状態遷移に追加されている。これらの新しい状態遷移は、DS(再同期化)Mismatchイベント、および(再)同期化終了(Done)25イベントと名付けることが可能である。交渉(フル)終了(Done)28、ローディング(フル)終了(Done)27、および交換(フル)終了(Done)26など他の追加の状態遷移は、本発明の新しい状態と共に使用される。これらの追加の状態遷移は、それぞれ、交渉終了(Done)15、ローディング終了(Done)21、および交換終了(Done)17の既存の状態遷移を反映している。これについては、表1を参照して、以下でより詳細に説明する。追加された状態と、それに関連する状態遷移を除いて、有限状態機械10を備える様々な状態は、従来、PNNI仕様のセクション5.7.2に定義されている。

【0028】通常PNNIプロトコルに存在し、PNNI仕様のセクション5.7.1に定義されている隣接ピアデータ構造も、既存のプロトコルと比較すると、追加のタイマと関連するタイマの間隔の定義によって、本発明により変更されている。追加のタイマと間隔は、それぞれ、記述の簡便化のために、再同期化非活動タイマおよび再同期化非活動間隔と名付けることが可能である。この新しい時間間隔が満了するとき、データベースの再同期化に関する潜在的な問題が信号で示されることになる。そのような場合、データベースの同期化が、既存のPNNIプロトコルにおいて当初行われた場合に通常起こるように、交渉状態14からデータベースの再同期化を実施するために、ノードの再同期化が要求される。

【0029】最後に、追加のビットが、本発明の一実施形態により、データベースサマリパケット構造に提供される。追加のビットについては、以下でより完全に説明

10

20

30

40

50

し、これは、フル状態での同期化（Synch in Full）（SF）ビットと名付けることが可能である。このデータベースサマリパケット構造に追加されたビットは、例えばルーティングエンティティの故障のために、再同期化が行われており、したがって、この再同期化が、既存のPNNIプロトコルにより続いて行われないことを、隣接ノードに信号で示すことを意図している。

【0030】図2を参照すると、知られている隣接ピア有限状態機械5を既存のPNNI仕様において使用して、所与のノードとその近隣の隣接ピアの各々との間で10 行われているデータベースの同期化の状況とフラグディングが記述されている。知られているように、隣接ピアに対して指定された有限状態機械5は、特定の隣接ピアへの活動リンクが存在しないことを示す初期状態12（名称「NPダウン」）を有する。隣接ピアで近接性を創出する第1ステップは、当業者に知られている交渉状態14である。初期状態12から交渉状態14への遷移をトリガするイベントは、追加ポートイベント13と名付けられている。ここで、隣接ピアへのリンクのための20 ハロー状態機械は、この分野に精通している者には知られているように、通常、両方向インサイド状態に到達している。交渉状態14では、2つの隣接ピアが、どちらのノードが、データベースを交換するためのマスタであるかを決定し、初期DSシーケンス番号が選択される。当業者なら、DSシーケンス番号を使用して、個々のデータベースサマリパケットを識別することを理解するであろう。

【0031】交渉状態14が完了した後、隣接ピアノードは、交換状態16中に、当該ノードから、データベースサマリパケットを受信する。したがって、交渉終了（Done）イベント15は、有限状態機械10を交換状態16に遷移させる。PNNIプロトコルにおいてすでに知られているように、交換状態16では、当該ノードは、隣接ノードへのトポロジデータベースを記述する。隣接ノードがデータベースサマリパケットを処理した後、必要とするPTSE情報の要求へと進むことができる。そのような情報が要求された場合、交換終了（Done）イベント17は、有限状態機械5をローディング状態18に遷移させる。

【0032】ローディング状態18中に、必要なPTSE情報は、隣接ノードによって要求され、少なくとも1つのPTSEヘッダは、まだそのノードによって受信されていない。知られているPNNIプロトコルでは最後に、2つのイベントのどちらかが生じた際に、フル状態20が達成される。まず、ローディング終了（Done）イベント21によって、ローディング状態18に続いて、PTSE情報の受信が完了した後、フル状態20に到達する。代替として、データベースサマリパケットの処理により、隣接ノードによってPTSE情報が要求

されないことが明らかになる場合、同期化終了（Done）イベント19によって、交換状態16に続いて、直接フル状態20に到達することが可能である。フル状態20では、隣接ピアから入手可能であることが知られているすべてのPTSE情報が、当該ノードによって所有され、その後、PTSEによって、隣接ピアへのリンクを通告することができる。

【0033】上述したように、特に図3を参照すると、本発明の一実施形態によって、新しい状態が知られている有限状態機械5に追加され、変更した有限状態機械10が達成されている。これらは、フル状態での交換（Exchange in Full State）22とフル状態でのローディング（Loading in Full State）24を含む。フル状態20のノードが、故障の前またはさらに同期化するためのいくつかの他の用件の前に、トポロジデータベースをすでに同期化し、ノードがその後データベースの再同期化を要求する場合、ノードは、初期化したデータベースサマリパケットを隣接ピアノードに送信する。上記で触れたように、これらのデータベースサマリパケットは、それぞれのフル状態での同期化（Synch in Full）（SF）ビットセットを有することになる。次いで、交渉（フル）終了（Done）イベント28は、有限状態機械10をフル状態での交換（Exchange in Full State）22に遷移させる。フル状態での交換（Exchange in Full State）22において、ノードは、フル状態20に到達する際に通告された隣接への局所状態情報の通告を中止せずに、隣接ノードにトポロジデータベースを記述することを試みる。同様に、当該ノードが同期化している隣接ノードも、データベースの再同期化を要求するノードに対する、それぞれの局所的な状態情報の通告を中止しない。データベースサマリパケットは、再同期化を要求しているノードの隣接ピアに送信される。これらのデータベースサマリパケットも、それぞれのフル状態での同期化（Synch in Full）（SF）ビットセットを有する。データベースサマリパケットを処理した結果、故障したノードが、PTSEパケットが必要でないと決定する場合、有限状態機械10は、（再）同期化終了（Done）イベント25によって、フル状態20に20 25 30 35 40 45 50 遷移する。

【0034】一方、PTSEパケットが必要な場合、有限状態機械10は、フル状態でのローディング（Loading in Full State）24に遷移するために、交換（フル）終了（Done）イベント26を実行する。次いで、必要なPTSEが、フル状態20の場合のように、依然として通告されている隣接ピアノードへのリンクとの再同期化を必要としているノードによって要求される。すべての要求されたPTSEパケットが、同期化を要求しているノードによって受信された

後、ローディング（フル）終了（Done）イベント27は、有限状態機械10をフル状態20に遷移させる。したがって、再同期化を要求しているノードは、そのピアデータ構造が、フル状態20、フル状態での交換（Exchange in Full State）22、およびフル状態でのローディング（Loading in Full State）24にある間、データベースサマリ、PTSE要求、およびPTSPパケットを送信および受信することになる。また、上述した同じ状態遷移が、同期化を要求している故障したノードの隣接ピアノードに関して行われる。

【0035】したがって、本発明により、2つのデータベース同期化手順が提供されており、一方は、ピアデータ構造がフル状態20にあるときに動作し、他方は、ピアデータ構造が交渉状態14にあるときに動作する。最低レベルの隣接ピア間のリンクは、隣接ピアの有限状態機械10が、フル状態20、フル状態での交換（Exchange in Full State）22、またはフル状態でのローディング（Loading in Full State）24にあるとき、PTSEにおいてのみ通告される可能性がある。したがって、物理リンクまたはVPCによって接続されている隣接する最低レベルのピアについて、フル状態での交換（Exchange in Full State）22またはフル状態でのローディング（Loading in Full State）24以外の状態からフル状態20への変化およびフル状態20、フル状態での交換（Exchange in Full State）22、およびフル状態でのローディング（Loading in Full State）24からなる状態のグループから

の変化により、同期化ノードまたは再同期化ノードに対する1つまたは複数のPTSEの新しい事例が、発信またはフラッシュされることになる。

【0036】図2に示した状態遷移のほかに、知られているPNNIプロトコルは、4つの追加のイベントを有する。これらは、DSMismatchイベント、BadPTSERequestイベント、DropPortLastイベント、およびDropPortLastイベントと名付けられている。DSMismatchイベントおよびBadPTSERequestイベントの各々は、状態を交渉状態14に遷移させる。DSMismatchは、データベースサマリパケットが、以下の現象のいずれかと共に受信されるときにはいつでも生じる。（i）予期しないDSシーケンス番号を有する、（ii）予期せずに設定された初期化ビットを有する、（iii）マスタビットの予期しないセッティングを有する。本発明によれば、知られているDSMismatchイベントは、データベースサマリパケットがフル状態での同期化（Synch in Full）ビットの

予期しないセッティングを有するときにもいつでも生じるように変更されている。そのうえ、本発明によれば、DSMismatchは、以前に記述したDS再同期化非活動タイマが満了するときにも生じる。以上の条件のいずれも、エラーがデータベース同期化プロセスに生じたことを示している。知られているDropPortLastイベントは、何ら状態の変化を生じず、知られているDropPortLastラストイベントは、NPダウン状態12を強制する。知られているDropPortLastイベントでは、隣接ピアへのリンクに対するハロー状態機械は、両方向インサイド状態を出すことになる。知られているDropPortLastラストイベントでは、隣接ノードへのすべてのポートはドロップされていることが決定される。

【0037】上記で導入したように、DS（再同期化）Mismatchイベントと名付けることが可能であるイベントは、データベースサマリパケットが、そのフル状態での同期化（Synch in Full）ビットが設定された状態で受信され、そのようなパケットが、予期されずに設定された初期化ビットを有する時にはいつでも行われるように定義されている。知られているPNNIプロトコルにおけるDSMismatchイベントの議論の場合のように、DSResynchMismatchイベントの出現は、同様に、エラーがデータベース再同期化のプロセスにおいて生じたことを示す。DSResynchMismatchは、データベースの再同期化が、有限状態機械10が交渉状態14にドロップせずに、再試行されることを示す。

【0038】次に、本発明の一実施形態による隣接ピア有限状態機械10について、以下に記述した表1を参照して、より詳細に説明する。表1では、新しいフル状態での交換（Exchange in Full State）22とフル状態でローディング状態24が、要約した形態で、これらの状態の潜在的な遷移状態と共に示されており、両方とも、知られているPNNIプロトコルに存在し、本発明によって、新しく追加または変更されている。表の各セルは、遷移イベントと指定された遷移イベントの開始時におけるノードの現在の状態として示された状態とのペアリングを表す。各セルは、指定された遷移イベントの結果として達成された新しい状態ならびに当該ノードによって取られたアクションを反映している。表1は、変更された有限状態機械10になるように、本発明によって変更された、または本発明を実施するために追加の機能または手順によって補足されたPNNIプロトコルの既存の隣接ピア有限状態機械5の態様のみを表示している。既存のPNNIプロトコルに対するこれらの変更と追加については、以下でさらに詳細に議論する。

【表1】

表1: 既存のPNNI隣接ピア有限状態機械に対する変更

遷移イベント	状態						
	NPダウン	交渉	交換	ローディング	フル	交換フル	ローディングフル
追加ポート	Ds1 交渉	—	—	—	—	Ds8 交換フル	Ds8 ローディングフル
交渉終了	—	—	—	—	—	FSM_ERR	FSM_ERR
交渉(フル)終了	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds11 交換フル	FSM_ERR	FSM_ERR
交換終了	—	—	—	—	—	FSM_ERR	FSM_ERR
交換(フル)終了	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds3 ローディングフル	FSM_ERR
同期化終了	—	—	—	—	—	FSM_ERR	FSM_ERR
(再)同期化終了	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds12 フル	FSM_ERR
ローディング終了	—	—	—	—	—	FSM_ERR	FSM_ERR
ローディング(フル)終了	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds12 フル
DS Mismatch	—	—	Ds5 交渉	Ds5 交渉	—	Ds6 交渉	Ds8 交渉
DS 再同期化 Mismatch	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds13 フル	Ds13 フル
Bad PTSE Request	—	—	Ds5 交渉	Ds5 交渉	—	Ds6 交渉	Ds8 交渉
Drop Port	—	—	—	—	—	Ds9 交換フル	Ds9 交換フル
Drop Portラスト	—	Ds10 NPダウン	Ds10 NPダウン	Ds10 NPダウン	Ds10 NPダウン	Ds10 NPダウン	Ds10 NPダウン

【0039】既存のPNNI仕様のセクション5. 7. 4の場合のように、FSM_ERRは、内部実施エラーを表す。したがって、イベント交渉(フル)終了(Done) 28は、有限状態機械10がフル状態20以外の状態の場合、通常生じない。同様に、イベント交換(フル)終了(Done) 26と(再)同期化終了(Done) 25は、フル状態での交換(Exchange in Full State) 22以外の状態中に生じない。そのうえ、ローディング(フル)終了(Done) イベント27は、フル状態でのローディング(Loading in Full State) 24を除くあらゆる状態において生じないことが予期される。さらに、DS(再同期化) Mismatchイベントは、フル状態での交換(Exchange in Full State) 22とフル状態でのローディング(Loading in Full State) 24を除くあらゆる状態において生じないことが予期される。

【0040】各既存のイベントである交渉終了(Done) 15、交換終了(Done) 17、同期化終了(Done) 19、およびローディング終了(Done) 21は、本発明の新しい状態である交換フル22とローディングフル24において生じるように準備されてい

い。この理由のために、プロトコルのエラー条件FSM_ERRが、前記既存のイベントを前記新しい状態にマッピングするために、表1の下で反映されている。既存のイベントは、以下でより完全に説明するように、類似のイベントである交渉(フル)終了(Done) 28、交換(フル)終了(Done) 26、(再)同期化終了(Done) 25、およびローディング(フル)終了(Done) 27によって置き換わっている。

【0041】当該ポートがNPダウン状態12にある間に、イベント追加ポートが行われる場合、PNNI仕様のセクション5. 7. 4においてDs1と名付けられているアクションに対し、PNNIプロトコルに記述されている知られている手順が続く。一般に、手順は、PTSEサマリを有さないデータベースサマリパケットを送信することを必要とする。これらのデータベースサマリパケットは、隣接ピアデータ構造の知られているDS Rxmt タイマによって指定された時間間隔で再送信される。本発明のDs1手順との違いは、データベースサマリパケットは、以前に記述したフル状態での同期化(Synch in Full) ビットの形態にある追加ビットを含むことである。表1で言及したDs1手順では、当該データベースサマリパケットのフルビットに

30

40

50

おける同期化は設定されていない。追加ポートイベントが、フル状態での交換 (Exchange in Full State) 22 中またはフル状態でのローディング (Loading in Full State) 24 中に行われる場合、状態機械は、同じ状態に留まることになる。物理リンクまたはVPCによって接続されている最低レベルの隣接ピアの場合、ポートIDが、隣接ピアデータ構造のポートIDリストに追加される。そのうえ、隣接ピアへのリンクが追加され、PTSEの新しい事例が発信される。このアクションのセットは、現

存するPNNIプロトコルに指定されたものと同じであり、PNNI仕様のセクション5.7.4では、アクションDs8として表されている。

【0042】有限状態機械10がフル状態20にあり、イベント交渉(フル)終了(Done)28が行われる場合、フル状態での交換 (Exchange in Full State) 22への遷移がトリガされ、表1でDs11と表されたアクションのセットが行われることになる。Ds11と名付けられたこのアクションのセットの下で、知られているPNNIプロトコルにおいてDs2と名付けられているアクションの場合のように、当該ノードが、トポロジデータベースのコンテンツのサマリを、データベースサマリパケットの形態にある隣接ピアに送信し始める。このフル状態での交換 (Exchange in Full State) 22への遷移に続く手順は、PNNI仕様のセクション5.7.4において、Ds2と表されているものと同一であるが、ノードによって送信されたデータベースサマリパケットが、フル状態での同期化 (Synch in Full) ビットセットを有する点が異なる。同様に、本発明によれば、再同期化非活動タイマは、このタイマがまだ動作していない場合、状態が遷移する際に、Ds2の手順の一部として開始されることになる。

【0043】交換(フル)終了(Done)26のイベントが、有限状態機械10のフル状態における交換22中に注入される場合、フル状態でのローディング (Loading in Full State) 24への遷移が行われることになる。その後、既存のPNNI仕様のセクション5.7.4に記述されているDs3アクションが続く。すなわち、DS Rxmt タイマが、以前に停止されていない場合、停止されることになる。当業者に知られているように、PTSEリクエストパケットが、当該隣接ピアあるいは他の隣接ピアに送信されるか、または送信され続ける。各PTSEリクエストパケットは、以前に発見されたが、まだ受信されていない、隣接ピアのより最近のいくつかのPTSEを要求する。これらのPTSEは、隣接ピアデータ構造において、PTSEリクエストのリストにリストされる。

【0044】(再)同期化終了(Done)イベント25が、状態機械がフル状態での交換 (Exchange

in Full State) 22にある間に注入される場合、有限状態機械10は、フル状態20に遷移することになる。フル状態での交換 (Exchange in Full State) 22に関連付けられているこのイベントは、交換状態16中に行われる同類物からこのイベントを区別するために、(再)同期化終了 (Done) イベント25と名付けられている。

【0045】ローディング(フル)終了(Done)イベント27が、有限状態機械10がフル状態でのローディング (Loading in Full State) 24にある間に注入される場合、状態機械は、フル状態20に遷移することになる。フル状態でのローディング (Loading in Full State) 24に関連付けられているこのイベントは、ローディング状態18中に行われる類似物からこのイベントを区別するために、図1では、ローディング(フル)終了 (Done) イベントと名付けられている。

【0046】ローディング(フル)終了(Done)イベント27がフル状態でのローディング (Loading in Full State) 24中に行われる状況、または(再)同期化終了(Done)イベント25がフル状態での交換 (Exchange in Full State) 22中に行われる状況では、本発明の例示的な実施形態において導入され、上述したDS Rxmtタイマと再同期化非活動タイマは、停止されることになる。当該状態の変化に続くこれらのアクションは、表1においてDs12と名付けられている。これは、データベースの再同期化が完了したことを表す。

【0047】DSMismatchイベントまたはBadPTSERequestイベントのどちらかが、交換状態16中またはローディング状態18中に注入される場合、交渉状態14への状態遷移が実施される。同様に、既存のPNNIプロトコルでDs5と表されている知られている手順が、当該状態遷移に続けて開始される。これらの手順は、PNNI仕様のセクション5.7.4に記述されている。しかし、本発明によれば、知られているプロトコルのピア遅延Ackタイマ、DS Rxmtタイマ、およびリクエスト Rxmtタイマの場合のように、再同期化非活動タイマも、以前に停止されていない場合、停止されることになる。本発明によれば、知られているDs5の手順で、ノードによって送信されたデータベースサマリパケットのフルビットにおける同期化は、設定されないことになる。

【0048】フル状態での交換 (Exchange in Full State) 22中またはフル状態でのローディング (Loading in Full State) 24中に生じる表1のイベントDSMismatchおよびBadPTSERequestは、交渉状態14に遷移することになる。行われるアクションは、交換状態中またはローディング状態中に生じる同じイベ

10

20

30

40

50

ントの場合上記で指定されているように、PTSEが、隣接へのリンクのあらゆる通告を排除するように変更される点異なる。後者のステップは、既存のPNNIプロトコルにおいて知られており、PNNI仕様のセクション5.7.4において、アクションDs6として記述されている。

【0049】有限状態機械10は、再同期化非活動タイマが満了するときにはいつでも、イベントDSMismatchで実行される。これは、ノードに、現在PTSEにおいて通告されている非応答隣接へのすべてのリンクを中止させる。次いで、有限状態機械10は、ノードが、再度、交渉状態14から開始する知られている方式で、データベースを同期化することを試みる場合、交渉状態14への遷移が強制される。データベースの再同期化が成功したイベントでは、再同期化非活動タイマが停止される。

【0050】さらに表1を参照すると、DS（再同期化）Mismatchが、フル状態での交換（Exchange in Full State）22中またはフル状態でのローディング（Loading in Full State）24中に生じるとき、フル状態に戻る。この遷移イベントの際に行われるアクションにより、ピア遅延Ackタイマ、DS Rxmtタイマ、およびリクエスト Rxmt タイマが、以前に停止されていない場合、停止されることになる。前記タイマのすべては、既存のPNNIプロトコルにおいて知られている。ピア再送信リスト、ピア遅延Ackリスト、PTSEリクエストリスト、およびすべての関係するタイマもクリアされる。フル状態での同期化（Synch in Full）ビットが設定されたデータベースサマリ 30の交換を、再度やり直さなければならない。次いで、当該ノードは、この隣接ピアのDSシーケンス番号を増加させて、マスタビットを値1に設定することによって、自身をマスタと宣言し、初期化ビット、モア（More）ビット、マスタビット、およびフル状態での同期化（Synch in Full）ビットが設定されたデータベースサマリパケットの送信を開始する。PTSEサマリは、これらのパケットに含まれていない。最後に、DS Rxmtタイマが始動され、データベースサマリパケットの受信を確認するDBサマリパケットが受信されない場合、各DS Rxmt 間隔時間ごとにデータベースサマリパケットが再送信される。状態がフル状態20に変化する際のこれらのアクションのすべては、表1ではDs13と名付けられている。

【0051】有限状態機械10が、それぞれフル状態での交換（Exchange in Full State）22とフル状態でのローディング（Loading in Full State）24の同じ状態にあれば、表1のDrop Port Lastイベントは、これらの状態を保有している有限状態機械10に帰着す 50

る。既存のPNNIプロトコルのセクション5.7.4のDs9手順のように、当該リンクは、対応する隣接ピアデータ構造のポートIDリストから削除されている。そのリンクを通告するPTSEが存在する場合、影響を受けたPTSEの新しい事例が発信されることになる。リンクが、隣接への最後の活動リンクである場合、Drop Port Lastラストイベントが生成されることになる。表1に示したように、Drop Port Lastラストイベントは、有限状態機械10が、フル状態での交換（Exchange in Full State）22またはフル状態でのローディング（Loading in Full State）24のどちらかにある場合にいつでも、有限状態機械10をNPダウン状態12に遷移させる。PNNIプロトコルにおいて知られており、PNNI仕様のセクション5.7.4においてDs10と名付けられているように、ピア遅延Ackタイマ、DS Rxmtタイマ、およびリクエストRxmtタイマは、以前に停止されていない場合、停止される。既存のDs10手順において知られているように、ピア送信リスト、ピア遅延Ackリスト、およびPTSEリクエストリストは、それらの関係するタイマと共にクリアされる。しかし、本発明によれば、再同期化非活動タイマも、以前に停止されていない場合、停止されることになる。

【0052】データベースサマリパケットの送信
データベースサマリパケットの送信は、以下に示した点を除いて、一般に、知られているPNNIプロトコルによって指定されている。しかし、フル状態20において、ノードは、設定された初期化ビット、モアビット、マスタビット、およびフル状態での同期化（Synch in Full）ビットを有する空のデータベースサマリパケットを送信することによって、データベースの再同期化を開始することが可能である。再同期化非活動タイマは、このタイマがまだ動作していない場合、第1のそのようなデータベースサマリパケットが送信されたときに始動される。そのようなデータベースサマリパケットを送信するとき、DS Rxmtタイマが再始動されなければならない。これらのパケットは、DS Rxmtタイマがスタートするとき、DS Rxmt 間隔秒ごとに、データベースの再同期化を開始するノードによって再送信される。

【0053】フル状態20のノードも、データベースの再同期化を要求している隣接ピアから、設定された初期化ビット、モアビット、マスタビット、およびフル状態での同期化（Synch in Full）ビットを有する受信したデータベースサマリパケットに応答して、データベースサマリパケットを送信する。ノードが、データベースの再同期化を開始するとき、値1に設定されている、フル状態での同期化（Synch in Full）ビット、初期化ビット、モアビット、およびマス 50

タビットを有するデータベースサマリパケットを送信する。データベースの再同期化を要求している隣接ノードに応答するノードは、以下でより完全に説明するように、フル状態での同期化(Synch in Full)ビット、初期化ビット、モアビット、およびマスタビットを設定する。ノードが、データベースの再同期化を要求している隣接ピアに対する第1のそのようなデータベースサマリパケットに応答するとき、ノードは、再同期化非活動タイマがまだ動作していない場合、それを始動させる。

【0054】交渉状態14において、当該ノードは、既存のプロトコルのように、空のデータベースサマリパケットを送信するが、フル状態での同期化(Synch in Full)ビットが、前記フル状態での同期化(Synch in Full)ビットの形態で追加のビットを含むように、本発明により変更されたデータベースサマリパケットに設定されない点が異なる。フル状態での交換(Exchange in Full State)22において、当該ノードは、知られている交換状態16におけるものと同一なデータベースサマリパケットを送信するが、フル状態での同期化(Synch in Full)ビットは、そのようなデータベースサマリパケットにおいて設定されている。

【0055】データベースサマリパケットの受信次に、ノードによる受信したデータベースサマリパケットの処理について説明する。データベースサマリパケットの受信は、一般に、知られているPNNIプロトコルによって指定されているが、以下に指摘した点が異なる。データベースサマリパケットが受け入れられる場合、多くのパケットフィールドは、ラスト受信データベースサマリパケットの識別情報として同定される知られている記録の対応する隣接ピアデータ構造に保存される。したがって、初期化ビット、モアビット、マスタビット、および予約ビットからなるパケットのフラグは、DSシーケンス番号と共に、すべて既存のPNNIプロトコルにおいて知られている方式で保存される。しかし、本発明のフル状態での同期化(Synch in Full)ビットも、以上のビット共に保存される。これらの保存された項目は、隣接ピアから受信された2つの連続データベースサマリパケットにおいて、全く同様に設定される場合、第2のそのようなデータベースサマリパケットは、受信したデータベースサマリパケットの処理において、重複であると見なされる。

【0056】知られているPNNIプロトコルの場合のように、データベースサマリパケットは、有限状態機械10がNPダウン状態12にある場合、無視されなければならない。

【0057】既存のPNNI仕様では、有限状態機械5が交渉状態14にあるときに受信され、ヘッディング「交渉」の下で、既存のPNNI仕様のセクション5、

7. 6において指定された2つの場合の一つと整合するデータベースサマリパケットは、イベント交渉終了(Done)を有し、交換状態16への遷移を有する有限状態機械5を実行する。知られているプロトコルのこれらの場合の第1では、初期化ビット、モアビット、およびマスタビットは、値1に設定されている。本発明では、上記の値を有する初期化ビット、モアビット、およびマスタビットの他に、シーケンスにおいて次として受け入れられ、セクション5. 7. 6のヘッディング「交渉」の下で議論した第1の場合によるすでに知られている方式でさらに処理するパケットに対して、フル状態での同期化(Synch in Full)ビットが、値ゼロに設定されなければならない。知られているプロトコルのこれらの場合の第2では、初期化とマスタのビットは、値ゼロに設定される。本発明では、上記の値を有する初期化とマスタのビットの他に、フル状態での同期化(Synch in Full)ビットが、シーケンスにおいて次として受け入れられ、セクション5. 7. 6のヘッディング「交渉」の下で議論した第2の場合によるすでに知られている方式でさらに処理されるパケットに対して、値ゼロに設定されなければならない。

【0058】有限状態機械10が交換状態16にあるときに受信されるデータベースサマリパケットに対して、セクション5. 7. 6のヘッディング「交換」の下で、既存のPNNI仕様において設定された手順が続くが、1つ例外がある。すなわち、セクション5. 7. 6の前記ヘッディングの下で列挙した第3の既存のステップの直後に、新しい条件が挿入される。この新しい条件は、フル状態での同期化(Synch in Full)ビットが設定される場合、イベントDSMismatchが生成され、データベースサマリパケットの処理が停止されることを要求する。交換状態16においてデータベースサマリパケットを受信することに続く手順は、他の場合は、既存のPNNI仕様に記述されている。

【0059】ノードがローディング状態18にあるときのデータベースサマリパケットの受信は、ノードが、データベースサマリパケットのシーケンス全体を送信および受信したことを表す。したがって、そのように受信されたパケットのみが、複製であるはずである。受信したあらゆる他のデータベースサマリパケットは、イベントDSMismatchを生成し、知られているPNNIプロトコルにより、再びデータベースを同期化する2つのピアを有する交渉状態14に戻らなければならない。ローディング状態18においてデータベースサマリパケットを受信するときに続く手順は、したがって、既存のPNNIプロトコルのセクション5. 7. 6に指定されている。

【0060】ノードがフル状態20にあり、受信したデータベースサマリパケットが設定されたフル状態での同期化(Synch in Full)ビットを有する場

10

20

30

40

50

合、受信したパケットが、それに続く2つの場合の一つと整合すれば、有限状態機械10は、交渉（フル）終了（Done）イベント28で実行されなければならない。それにより、有限状態機械10は、フル状態での交換（Exchange in Full State）22に遷移する。その後、パケットは、シーケンスにおいて次として受け入れられ、以下で記述するように、さらに処理される。

【0061】ケース1：受信ノードがスレープである
このケースは、初期化ビット、モアビット、およびマスタのビットが値1に設定され、パケットのコンテンツが空であり、隣接ピアノードIDが、受信ノードに関連付けられているノードIDより大きい状況を扱っている。

【0062】この状況では、当該ノードがスレープであり、交渉（フル）終了（Done）イベント28を生成すると、スレープノードは、以下のシーケンスのアクションを取る。まず、DS Rxmtタイマを停止する。次に、再同期化非活動タイマがまだ動作していない場合、それを始動させる。次いで、マスタビットを値ゼロに設定し（所与のノードがスレープであることを示す）、初期化ビットも値ゼロに設定し、フル状態での同期化（Synch in Full）ビットを値1に設定し、DSシーケンス番号をマスタノードによって指定された値に設定し、データベースサマリパケットは、既存のPNNI仕様のセクション5.7.5に記述されているように、当該ノードに関するデータベースサマリ情報のその部分を含めて、マスタに送信される。

【0063】ケース2：受信ノードがマスタである
このケースは、初期化ビットとマスタビットが値ゼロに設定され、パケットのDSシーケンス番号が当該ノードのDSシーケンス番号に等しく（確認を示す）、隣接ピアのノードIDが、所与のノードのものより小さい状況を扱っている。

【0064】このケースでは、当該ノードがマスタノードである。交渉（フル）終了イベント28を生成すると、マスタノードは、以下のアクションのシーケンスを取る。まず、DS Rxmtタイマを停止しなければならない。次いで、受信したデータベースサマリパケットのコンテンツを、受信したと確認しなければならない。その後、これらのコンテンツを、ヘッディング「交換状態」の下で、PNNI仕様のセクション5.7.6において、94ページと95ページで説明しているように、既存のPNNIプロトコルにおいて知られている方式で処理する。DSシーケンス番号は、値1だけ増加されなければならない。フル状態での同期化（Synch in Full）ビットは、値1に設定され、初期化ビットは、値ゼロに設定されなければならない。データベースサマリパケットは、既存のPNNI仕様のセクション5.7.5に記述されているように、当該ノードに関するデータベースサマリ情報のその部分を含めて、スレ

ブノードに送信されなければならない。DS Rxmtタイマが再始動されなければならない。再同期化非活動タイマは、まだ動作していない場合、始動される。

【0065】上記のケース1または2のどちらの適用可能でない場合、当該パケットは、それが複製であるか否かについて確認される。パケットが複製である場合、それは無視される。そうではなく、パケットが複製でない場合、エラーが再同期化中に生じており、その後、イベントDSMismatchで有限状態機械10を実行しなければならない。

【0066】フル状態20のノードと受信したデータベースサマリパケットが、設定されたフル状態での同期化（Synch in Full）ビットを有さない場合、当該パケットは、複製であることが予期される。設定されていないフル状態での同期化（Synch in Full）ビットを有するあらゆる他のデータベースサマリパケットは、イベントDSMismatchを生成し、それにより、有限状態機械10を交渉状態14に戻し、当該2つの隣接ピアにそれらのデータベースを再同期化させる。本発明により、設定されていないフル状態での同期化（Synch in Full）ビットを有するデータベースサマリパケットを受信する際に続く手順は、知られている交換状態16において続く手順と同じであるが、シーケンスにおいて次として受け入れられたパケットが、イベントDSMismatchを生成しなければならない。そのようなパケットをさらに処理することが停止される点が異なる。矛盾するマスタビットまたは値ゼロに設定された初期化ビットを有するパケットの受信も、イベントDSMismatchを生成しなければならない。

【0067】ノードの状態がフル状態での交換（Exchange in Full State）22である場合、受信したデータベースサマリパケットは、以下の条件付きステップを実行することによって、本発明により処理されることになる。これらの条件付きステップのいずれかが真であるとテストされた場合、残りのステップをテストまたは実行する必要はない。

【0068】1）データベースサマリパケットが、マスタとして作用するノードによって受信される場合、パケットの処理は、パケットが複製であると判定されれば、停止される。複製パケットの判定については、以前に議論した。

【0069】2）一方、ノードがスレープとして作用する場合、複製パケットの受信に応答して、マスタに送信された最後のデータベースサマリパケットを再送信し、次いで、受信したデータベースサマリパケットの処理を停止する。

【0070】3）パケットがあらゆるイベントにおいて複製でなく、マスタビットの状態が、接続のマスタ/スレープ状態と矛盾する場合、イベントDSMismatch

10

20

30

40

50

chが生成され、パケットの処理が停止される。

【0071】4) マスタビットは一貫しているが、フル状態での同期化(Synch in Full)ビットが設定されていない場合、イベントDSMismatchが生成され、パケットの処理は停止される。

【0072】5) 上記に列挙した条件のすべてが偽であるとテストされ、初期化ビットが設定されている場合、イベントDS Resynch Mismatchが生成され、パケットの処理が阻止される。

【0073】6) 上記に列挙した条件のすべてが偽であるとテストされ、ノードがマスタである場合、パケットのDSシーケンス番号がノードのDSシーケンス番号に等しいときに、パケットは受け入れられ、さらに処理される。これは、パケットがシーケンスにおいて次であることを示す。この場合の処理は、以下の方式で実施される。

【0074】DS Rxmtタイマが停止される。これに続いて、最も最近受信したデータベースサマリパケットのコンテンツが受信されたことを確認し、次いで、そのコンテンツを、ヘッディング「交換」の下で、PNNI仕様のセクション5.7.6において、94ページと95ページで説明されているように、既存のPNNIプロトコルにおいて知られている方式で処理する。次いで、DSシーケンス番号を値1だけ増加させる。次に、ノードがすでに、データベースサマリパケットのシーケンス全体を送信し、受信したパケットが、値ゼロに設定されているモアビットを有する場合、PTSEリクエストリストが空であれば、イベント(再)同期化終了(Done)25が生成され、PTSEリクエストリストが空でなければ、イベント交換(フル)終了(Done)26が生成される。ノードがデータベースサマリパケットのシーケンス全体を送信したか否かについての判定は、以前にノードによって送信されたデータベースサマリパケットも、値ゼロに設定されているモアビットを有する場合、行われる。データベースサマリパケットのシーケンス全体がまだ受信されていない場合、新しいデータベースサマリパケットがスレーブに送信され、DS Rxmtタイマが再始動される。

【0075】7) 上記に列挙した条件のすべてが偽であるとテストされ、ノードがスレーブである場合、パケットのDSシーケンス番号が、ノードのDSシーケンス番号より大きいときに、パケットは受け入れられ、さらに処理される。これは、パケットがシーケンスにおいて次であることを示す。この状況でのパケットの処理は、以下のよう、2ステップの手順による。

【0076】第1ステップとして、受信されたと認識されたデータベースサマリパケットのコンテンツは、次いで、ヘッディング「交換」の下で、PNNI仕様のセクション5.7.6において、94ページと95ページで説明しているように、既存のPNNIプロトコルにお

て知られている方式で処理される。第2ステップとして、以下のアクションが実施される。DSシーケンス番号は、受信したパケットに現れるDSシーケンス番号に設定される。次いで、データベースサマリパケットがマスタに送信される。次に、受信したパケットが、ゼロに設定されているモアビットを有し、送信されたばかりのデータベースサマリパケットが空であり、したがって、それも、ゼロに設定されたモアビットを有する場合、PTSEリクエストリストが空であれば、イベント(再)同期化終了(Done)25が生成される。PTSEリクエストリストが空でない場合、イベント交換(フル)終了(Done)26が実行される。

【0077】8) 上記に列挙した条件のすべてが偽であるとテストされる場合、イベントDS Matchが生成され、データベースサマリパケットの処理が停止される。

【0078】フル状態でのローディング>Loading in Full State)24に関して、この状態のノードは、設定されたフル状態での同期化(Synch in Full)ビットを有するデータベースサマリパケットのシーケンス全体を送信および受信することになる。したがって、この状態のノードによって受信されたデータベースサマリパケットのみが、複製であるはずである。再同期化を要求していない受信されたあらゆる他のデータベースサマリパケットは、イベントDSMismatchを生成しなければならない。これにより、交渉状態14に戻り、次いで、当該2つの隣接ピアは、知られている手順によるデータベースの同期化に進むことになる。フル状態でのローディング>Loading in Full State)24においてデータベースサマリパケットを受信するときに続く手順は、フル状態での交換(Exchange in Full State)22において続く手順と同じであるが、シーケンスにおいて次として受け入れられたパケットが、代わりにイベントDSMismatchを生成しなければならない、そのようなパケットのさらなる処理が停止される点異なる。矛盾するマスタビットを有するパケットの受信も、イベントDSMismatchを生成することになる。唯一矛盾するビットとして初期化ビットを有するあらゆるパケットは、イベントDS再同期化ミスマッチを生成することになる。

【0079】上述したように、追加のフラグが、既存のPNNIプロトコルの別の場合では知られているデータベースサマリパケットに対し、本発明により定義される。例えば、データベースサマリパケットの知られているフラグフィールドにおけるビット番号13を使用して、簡便のために、フル状態での同期化(Synch in Full)(SF)ビットと名付けることが可能である、この新しいフラグを定義することが可能である。このビットは、この交換プロセスを、既存のPNN

Iプロトコルにおいて知られている同期化から区別するために、データベースの再同期化を実施する2つのノードの各々によって、値1に設定される。

【0080】当業者なら、ある場合には、トポロジ状態ルーティングプロトコルを有するネットワークを、データベースの同期化の手順に予め不可欠である他のプロトコルと関連付けることが可能であることを理解するであろう。例えば、既存PNNIプロトコルの特定の場合では、本発明を適用する場合に、ハロープロトコルが、活動ルーティングエンティティから非活動ルーティングエンティティへの活動切替えのために、混乱されないことが保証されなければならない。保証されない場合、故障した活動ルーティングエンティティに関連付けられているネットワークノードは、活動切替えのためにのみ、到達不可能であると宣言されることになる。したがって、PNNIプロトコルなどのネットワークプロトコルの場合、活動ルーティングエンティティに関連付けられている局所的な状態情報は、非活動ルーティングエンティティに提供して、各々は、故障からの回復中に、局所的な状況情報の共通認識を共有するようにしなければならない。これは、例えば、当該故障の前に、活動ルーティングエンティティから非活動ルーティングエンティティに、周期的に局所的な状態情報を送信することによって達成することが可能である。

【0081】次に、本発明について、ネットワークドメイン30および55を備えるネットワーク2（図1）などの通信ネットワークにおけるホット冗長ネットワーク要素56に関して、図4を参照して、説明する。ネットワーク要素56も、ノード36などの論理グループノードを実施することが可能である。ネットワーク要素56は、入力/出力ポート100を備え、それを経て、ネットワーク接続76は、交換装置71によって経路指定される。ネットワーク接続76aは、76bにおいて、活動ルーティングエンティティ68の故障の際に、活動ルーティングエンティティ68から非活動ルーティングエンティティ70に、再経路指定または転換することが可能である。活動ルーティングエンティティ68と非活動ルーティングエンティティ70の両方とも、ネットワークドメイン30と55を有する通信ネットワーク2に関するトポロジ状態情報へのアクセスを備える。例えば、当該トポロジ状態情報は、それぞれトポロジデータベース74aと74bによって、活動ルーティングエンティティ68と非活動ルーティングエンティティ70にとってアクセスすることが可能である。

【0082】本発明によれば、ノード56などのホット冗長ネットワーク要素の活動ルーティングエンティティ68と非活動ルーティングエンティティ70の両方とも、活動ルーティングエンティティ68に影響を与える故障が発生する前に、所定の間隔で行われる、活動ルーティングエンティティ68から非活動ルーティングエン

ティティ70へのトポロジ状態情報の周期的な送信69によって、互いにトポロジ状態情報を共有することが可能である。代替として、そのような送信は、新しいデータが、イベント駆動方式で、活動ルーティングエンティティ68によって受信される際に行うことが可能である。同様に、非活動ルーティングエンティティ70は、代わりに、外部ソースの送信69aなどを介して、ノード56の外部から供給されたトポロジ状態情報で周期的に更新することが可能である。当業者なら、トポロジ状態情報で周期的に更新されていない非活動ルーティングエンティティ70を有するノード56と共に使用するために、本発明の方法を適応することも可能であることを理解するであろう。また、この分野に精通している者なら、ネットワーク要素56の活動ルーティングエンティティ68と非活動ルーティングエンティティ70の間でトポロジ情報を共有すること、ならびに、非活動ルーティングエンティティ70のトポロジ状態情報を周期的に更新するための機構は、知られている技術により実施することが可能であることを理解するであろう。

【0083】ホット冗長性技術の分野で知られているように、非活動ルーティングエンティティ70のトポロジ状態情報を周期的に更新することが望ましいだけでなく、同様の方式で、局所的な状態情報も、非活動ルーティングエンティティ70に周期的に送信することが可能である。例えば、PNNIプロトコルでは、そのような局所的な状態情報は、ハロープロトコルによって送信される。以前説明したように、これは、ネットワークノードが、データベースを再同期化する必要性を創出した活動切替えの理由によってのみ、ルーティングのために、故障したノードが到達不可能であると宣言することを防止する。したがって、回復したノードは、該当する活動切替えに続く局所的な状態情報を保有することになる。そのような局所的な状態情報は、同様の方式で、トポロジ状態情報に関して以前に記述したものに送信される。

【0084】ノード56（およびそれが実施する論理ノード36）が、隣接ピアと以前に同期化したトポロジデータベースを有し、これらのデータベースを再同期化することを望むとき、活動ルーティングエンティティ68に取って局所的である故障から回復するか、またはそのリセットを実施するかに関わらず、本発明を実施することにより、ノードは、局所的な状態情報の通告を終了させることにならない隣接ノードと、トポロジデータベース交換プロセスを開始することが可能になるが、これは、局所的な状態情報が、活動ルーティングエンティティ68の故障に関連するノードに関係し、かつ活動ルーティングエンティティ68の故障に関連するノードに直接隣接する各ノードに関係しているからである。そのような直接隣接ノードは、この用語が本明細書で使用されているように、物理的な隣接ノードと論理的な隣接ノードの両方を含む。例えば、PNNIプロトコルでは、活

動ルーティングエンティティ68の故障からの回復に含まれている様々なノードの間におけるリンクの通告は、データベースの交換中に終了されることはない。したがって、本発明によれば、活動ルーティングエンティティ68の故障に関連するノードは、局所的な状態情報の通告を中止しない。そのうえ、本発明によれば、活動ルーティングエンティティ68の故障に関連するノードに直接隣接する隣接ノードは、同様に、各そのような隣接ノードに関するそれぞれの局所的な状態情報の通告を中止することはない。

【0085】上述した本発明の方法は、ハードウェア、ソフトウェア、またはその組合せで実現することが可能である、データベース同期化プロセッサ78aと78bによって、それぞれ、活動ルーティングエンティティ68と非活動ルーティングエンティティ70の各々において実施することが可能である。同期化プロセッサ78aと78bは、ノード56とその直接隣接するより低いレベルの隣接57、59、60の各々との間と、ノード36とその直接隣接するより高いレベルの論理隣接34、35、37、38との間における、トポロジ状態情報の交換に関して以前に記述した様々な手順を実施する。上述したものは、活動ルーティングエンティティ68と非活動ルーティングエンティティ70の各々を有するそれぞれの同期化プロセッサ78aまたは78bを準備するが、当業者なら、単一の同期化プロセッサも、両方のデバイスによって共有することが可能であることを理解するであろう。

【0086】したがって、本発明は、ノードとそれに直接隣接する物理的または論理的な隣接ノードとの間で、同期化プロセス中に、これらのノードに、局所的な状態情報の通告を変更させない、データベースの再同期化機構を定義する。これにより、該当するノード間の同期化が、ネットワークのすべての他のノード、すなわち、故障したノードの直接隣接する物理的または論理的な隣接ノードでないノードにトランスペアレントとなることが保証されるはずである。後者の隣接ノードは、再同期化に携わることを要求されることが必要である唯一のノードである。

【0087】したがって、本発明により、ホット冗長性回復中に生じる可能性がある、データベースの再同期化のための知られている従来のシステムに関するある欠点に対処する試みが実施される。故障したノードが、特別の責務を所有する活動ルーティングエンティティである場合、活動ルーティングエンティティの故障は、活動ルーティングエンティティの特別の責務が、回復中に放棄される必要がないという意味で、一般に、ルーティングシステムまたはネットワークの混乱ではない。したがって、他のノードは、故障したノードの特別の責務を荷う必要はない。例えば、OSPFプロトコルのバックアップ指定ルータが、指定ルータとなることなく、新しい

ピアグループリーダーが、故障した特別のネットワーク構成要素を置換として、PNNIプロトコルの下の、ネットワーク階層のいかなるレベルにおいても、選択されることはない。

【0088】本発明における故障の検出と故障の回復は、故障したノード内と、物理的または論理的に直接隣接するノードまたはデバイスに関してのみ行われるので、本発明による回復プロセスは、ホット冗長性回復のための従来の技術のプロセスより、時間がかからないことが予期される。そのうえ、故障したネットワークエンティティに隣接する上述したものの以外のネットワークのデバイスが含まれないことが予期される場合、従来の技術またはデバイスの場合より、狭い範囲のネットワークトポロジの部分が回復に携わっている。回復に関する時間とトポロジの広がりについて予期されるこの利点は、故障したノードが通常特別の責務を実施しているか否かに関わらず、適用可能である。最後に、本発明の回復プロセスに関連するネットワークの帯域幅と処理要件は、故障した構成要素に直接隣接するノードのみが回復に含まれという点で、故障回復の知られている技術に対し、低減されることが予期される。

【0089】本発明について、既存のPNNIプロトコルに適応した実施に関して説明してきたが、通信ネットワークの分野の技術者なら、本発明は、他のトポロジ状態のプロトコルに適用または適応することが可能であることを理解するであろう。そのうえ、同期化プロセッサを本発明の方法を実行するために使用することが可能であると同時に、方法の様々なステップを、ネットワーク要素に関連する1つまたは複数の他のデバイスあるいはソフトウェアによって、全体または一部として実施することが可能であるか、または、ネットワーク要素に関連付けられ、本発明と共に使用するために修正または適応することが可能である既存のデバイスまたはソフトウェアによって実施することが可能である。さらに、本発明について、ホット冗長性回復の状況で説明してきたが、本発明は、故障回復の他の状況において、または、ノードが以前にデータベースの同期化を実施した後に、そのトポロジデータベースを再同期化することが望ましい時にはいつでも、使用することが可能である。さらに、当業者なら、詳細な様々な変更を本発明に行うことが可能であり、そのすべてが、本発明の精神と範囲内にあることを理解するであろう。

【図面の簡単な説明】

【図1】本発明の方法と装置を実施することが可能である、PNNIルーティングプロトコルにより動作するネットワークドメインに関連付けられており、ネットワークトポロジの一部を形成するノードのグループ間における親子関係を示す、階層的なネットワークトポロジの概略図である。

【図2】従来の技術で知られているPNNIルーティン

グプロトコルの隣接ピア有限状態機械に関する様々な状態と遷移イベントを示す状態機械の図である。

【図3】本発明を実施するように変更されたPNNIルーティングプロトコルの隣接ピア有限状態機械に関する様々な状態と遷移イベントを示す状態機械の図である。

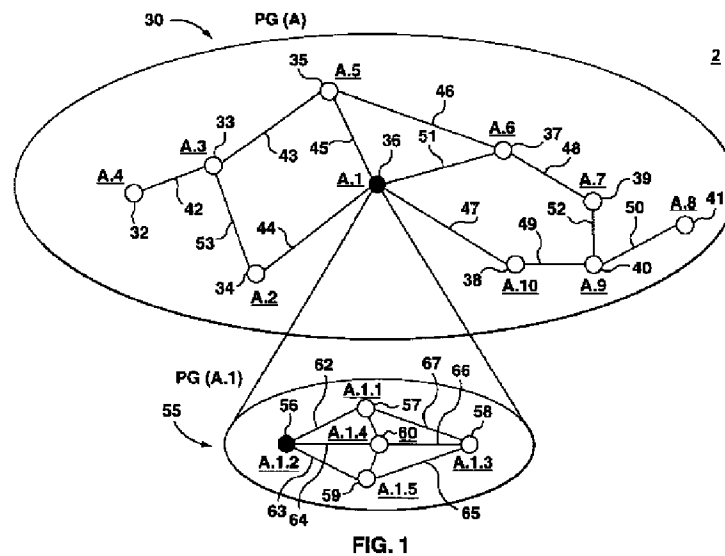
【図4】本発明の方法を実施することが可能であるホット冗長ネットワーク要素のブロック図である。

【符号の説明】

2 通信ネットワーク
5 隣接ピアノ有限状態機械
10 変更した隣接ピアの有限状態機械
12 初期状態 (NPダウン状態)
13 追加ポートイベント
14 交渉状態
15 交渉終了 (Done)
16 交換状態
17 交換終了 (Done) 状態
18 ローディング状態
19 同期化終了 (Done) イベント
20 フル状態
21 ローディング終了 (Done)
22 フル状態での交換 (Exchange in Full State)
24 フル状態でのローディング (Loading in Full State)

* n Full State)
25 (再) 同期化終了 (Done)
26 交換 (フル) 終了 (Done)
27 ローディング (フル) 終了 (Done)
28 交渉 (フル) 終了 (Done)
30 PNNIネットワークドメイン
32、33、34、35、36、37、38、39、40、41 ネットワークノード
36 ノード
42、43、44、45、46、47、48、49、50、51、52、53 物理ノードまたは論理ノード
55 子ピアグループ
56、57、58、59、60 より低いレベルのネットワークノード
62、63、64、65、66、67 物理リンクまたは論理リンク
68 活動ルーティングエンティティ
69、69a 周期的な送信
70 非活動ルーティングエンティティ
71 交換装置
74a、74b トポロジデータベース
76、76a、76b ネットワーク接続
100 入力/出力ポート

【図1】



【図2】

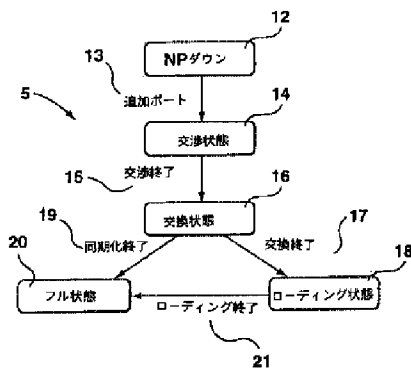


FIG. 2

【図3】

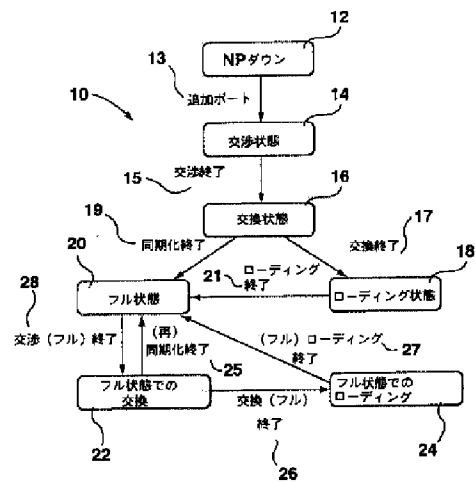


FIG. 3

【図4】

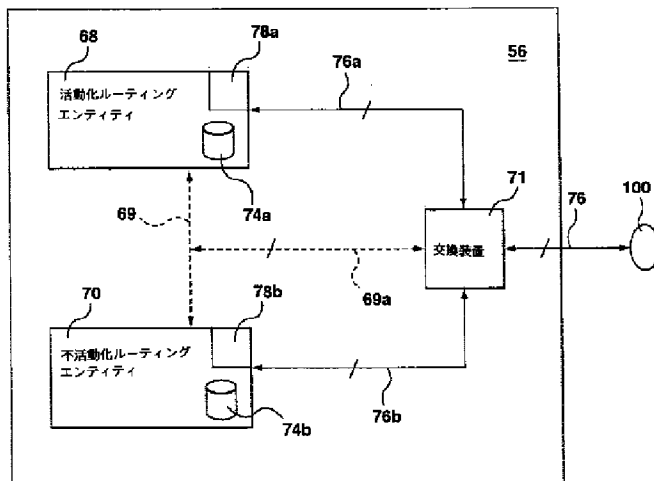


FIG. 4

フロントページの続き

(72)発明者 カール・ラジエシツク
カナダ国、オンタリオ・ケー・２・ビー・
７・テイ・１、ネピアン、ウツドリツジ・
クレツセント・１２２２－９０

Fターム(参考) 5K030 GA11 HA08 HC01 HC20 HD03
JL07 KA05 KA07 LA15 LB05
MA01 MD07 MD10

【外国語明細書】

1. Title of Invention
METHOD AND APPARATUS FOR TOPOLOGY DATABASE
RE-SYNCHRONIZATION IN COMMUNICATIONS NETWORKS
HAVING TOPOLOGY STATE ROUTING PROTOCOLS

2. Claims

1. A method for recovery from a failure which affects an active routing entity in a communications network, the active routing entity being associated with a network node of the communications network, the communications network comprising a routing protocol for intermittent advertisement of local state information throughout the network and further comprising an inactive routing entity to which network connections of the network node can be diverted from the active routing entity upon the said failure, the method comprising the steps of:

(a) upon the failure, executing an activity switch between the active routing entity and the inactive routing entity, wherein network connections of the network node are diverted from the active routing entity to the inactive routing entity to thereby transform the inactive routing entity into a newly active routing entity;

(b) following the activity switch, exchanging topology state information between the newly active routing entity and each immediately adjacent neighbour node of the network node associated with said failure such that the newly active routing entity and every said immediately adjacent neighbour node respectively possess synchronized topology state information; and

wherein the exchange of topology state information between the newly active routing entity and each said immediately adjacent neighbour node is performed without withdrawal, by the network node associated with said failure and by each said immediately adjacent neighbour node, of the said intermittent advertisement of local state information as it pertains respectively to the network node associated with said failure and to each said immediately adjacent neighbour node.

2. The method according to Claim 1, further comprising the step of transmitting topology state information to the inactive routing entity prior to the failure, whereby the active routing entity and the inactive routing entity both share a common understanding of overall network topology status immediately following said transmission of topology state information.

3. The method according to Claim 2, wherein the transmission of topology state information is periodic.
4. The method according to Claim 3, wherein the periodic transmission of topology state information to the inactive routing entity is from the active routing entity.
5. The method according to Claim 2, further comprising the step of transmitting local state information to the inactive routing entity prior to the failure, whereby the active routing entity and the inactive routing entity both share a common understanding of local status immediately following said transmission of local state information.
6. The method according to Claim 5, wherein the transmission of local state information is periodic.
7. The method according to Claim 6, wherein the periodic transmission of local state information to the inactive routing entity is from the active routing entity.
8. The method according to Claim 5, wherein the local state information comprises local link status information and local nodal status information.
9. The method according to Claim 8, wherein the local link status information is selected from the group comprising link characteristics, link operational status and port identifiers.
10. The method according to Claim 8, wherein the local nodal status information is selected from the group comprising node identifier, peer group identifier, distinguished node election status, distinguished node leadership status and local reachable addresses.

11. The method according to Claim 7, wherein the active routing entity and the inactive routing entity each forms part of the network node associated with said failure.
12. The method according to Claim 11, wherein the active routing entity and the inactive routing entity are each implemented by way of distinct physical components.
13. The method according to Claim 5, wherein the communications network is an Asynchronous Transfer Mode (ATM) network and the routing protocol for intermittent advertisement of local state information throughout the communications network is the PNNI protocol.
14. The method according to Claim 5, wherein the communications network is an Internet Protocol (IP) network and the routing protocol for intermittent advertisement of local state information throughout the communications network is the Open Shortest Path First (OSPF) protocol.
15. The method according to Claim 13, wherein topology state information transmitted from the active routing entity prior to the failure is extracted from a topology database associated with the active routing entity, and wherein topology state information which is exchanged following the activity switch between the newly active routing entity and every said immediately adjacent neighbour node is extracted from a topology database respectively associated with the newly active routing entity and every said immediately adjacent neighbour node.
16. The method according to Claim 15, wherein topology state information is transmitted from the active routing entity to the inactive routing entity prior to the failure by bundling the topology state information into PNNI Topology State Elements (PTSE).
17. The method according to Claim 15, wherein topology state information is exchanged following the activity switch between the newly active routing entity and

every said immediately adjacent neighbour node by bundling the topology state information into PNNI Topology State Elements (PTSE).

18. The method according to Claim 16, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP) for said transmission.

19. The method according to Claim 17, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP) for said exchange.

20. The method according to Claim 15, wherein the newly active routing entity, prior to the said exchange of topology state information with each said immediately adjacent neighbour node, notifies each said immediately adjacent neighbour node that the said exchange is to take place without said withdrawal of the said intermittent advertisement of local state information.

21. The method according to Claim 20, wherein the notifying of the said exchange of topology state information without said withdrawal of the said intermittent advertisement of local state information takes place by way of a flag in a notification message sent to each immediately adjacent neighbour node of the network node associated with said failure.

22. The method according to Claim 21, wherein the notification message is a PNNI Database Summary packet in which said flag is provisioned.

23. A network element for recovery from a failure in a communications network which includes a routing protocol for intermittent advertisement of local state information throughout the network, the network element comprising:

an active routing entity, wherein the active routing entity is associated with topology state information concerning the communications network;

an inactive routing entity, wherein an activity switch is executed between the active routing entity and the inactive routing entity upon failure of the active routing entity to thereby divert network connections from the active routing entity to the inactive routing entity and transform the inactive routing entity into a newly active routing entity;

a database synchronization processor, wherein the database synchronization processor effects an exchange of topology state information between the newly active routing entity and each immediately adjacent neighbour node of the network element following the activity switch such that the newly active routing entity and every said immediately adjacent neighbour node respectively possess synchronized topology state information, and wherein the said exchange of topology state information between the newly active routing entity and each said immediately adjacent neighbour node is performed without withdrawal, by the network node associated with said failure and by each said immediately adjacent neighbour node, of the said intermittent advertisement of local state information as it pertains respectively to the network element and to each immediately adjacent neighbour node.

24. The network element according to Claim 23, wherein the topology state information is transmitted to the inactive routing entity prior to the failure of the active routing entity, such that the active routing entity and the inactive routing entity both share a common understanding of overall network topology status following said transmission of topology state information.

25. The network element according to Claim 24, wherein the transmission of topology state information is periodic.

26. The network element according to Claim 25, wherein the periodic transmission of topology state information to the inactive routing entity is from the active routing entity.

27. The network element according to Claim 24, wherein local state information is transmitted to the inactive routing entity prior to the failure of the active routing entity,

whereby the active routing entity and the inactive routing entity both share a common understanding of local status immediately following said transmission of local state information.

28. The network element according to Claim 27, wherein the transmission of local state information is periodic.

29. The network element according to Claim 28, wherein the periodic transmission of local state information to the inactive routing entity is from the active routing entity.

30. The network element according to Claim 24, wherein the local state information comprises link status information and local nodal status information.

31. The network element according to Claim 30, wherein the local link status information is selected from the group comprising link characteristics, link operational status and port identifiers.

32. The network element according to Claim 31, wherein the local nodal status information is selected from the group comprising node identifier, peer group identifier, distinguished node election status, distinguished node leadership status and local reachable addresses.

33. The network element according to Claim 32, wherein the active routing entity and the inactive routing entity are each implemented by way of distinct physical components.

34. The network element according to Claim 26, wherein the communications network is an Asynchronous Transfer Mode (ATM) network and the routing protocol for intermittent advertisement of local state information throughout the communications network is the PNNI protocol.

35. The network element according to Claim 26, wherein the communications network is an Internet Protocol (IP) network and the routing protocol for intermittent advertisement of link status information throughout the communications network is the Open Shortest Path First (OSPF) protocol.
36. The network element according to Claim 34, wherein topology state information transmitted from the active routing entity prior to the failure of the active routing entity is extracted from a topology database associated with the active routing entity, and wherein topology state information which is exchanged following the activity switch between the newly active routing entity and every said immediately adjacent neighbour node is extracted from a topology database respectively associated with the newly active routing entity and every said immediately adjacent neighbour node.
37. The network element according to Claim 23, wherein topology state information is transmitted from the active routing entity to the inactive routing entity prior to the failure by bundling the topology state information into PNNI Topology State Elements (PTSE).
38. The network element according to Claim 23, wherein topology state information is exchanged following the activity switch between the newly active routing entity and every said immediately adjacent neighbour by bundling the topology state information into PNNI Topology State Elements (PTSE).
39. The network element according to Claim 37, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP) for said transmission.
40. The network element according to Claim 38, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP) for said exchange.

41. The network element according to Claim 23, wherein the newly active routing entity, prior to the said exchange of topology state information with each said immediately adjacent neighbour node, notifies each said immediately adjacent neighbour node that the said exchange is to take place without said withdrawal of the said intermittent advertisement of local state information.
42. The network element according to Claim 41, wherein the notifying of the said exchange of topology state information without said withdrawal of the said intermittent advertisement of local state information takes place by way of a flag in a notification message sent to each immediately adjacent neighbour node of the network element.
43. The network element according to Claim 42, wherein the notification message is a PNNI Database Summary packet in which said flag is provisioned.
44. A method for synchronization of topology state information between two network nodes in a communications network, the communications network comprising a routing protocol for intermittent advertisement of local state information throughout the network, the two network nodes comprising a requesting node which initiates a request for topology state synchronization and a replying node which receives said request and which communicates with the requesting node to provide topology state information to the requesting node which is not possessed by the requesting node when same initiates said request, the method comprising the step of selecting, prior to said request being made by the requesting node to the replying node, between a first and a second mode of synchronization, the first said mode providing for topology state synchronization which entails withdrawal, by the said requesting node and by the said replying node, of the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node, and the second said mode providing for topology state synchronization which maintains the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node.

45. The method according to Claim 44, further comprising the step of exchanging topology state information between the requesting node and the replying node once the request for topology state synchronization has been made by the requesting node to the replying node.
46. The method according to Claim 45, wherein prior to the said exchange of topology state information, the requesting node notifies the replying node that the said exchange is to take place according to one of the said first mode and the said second mode of synchronization.
47. The method according to Claim 46, wherein said notification takes place by way of a flag in a notification message sent by the requesting node to the replying node.
48. The method according to Claim 47, wherein the topology state information which is exchanged between the requesting node and the replying node is extracted from topology state databases each respectively associated with the said requesting and replying nodes.
49. The method according to Claim 47, wherein the communications network is an Asynchronous Transfer Mode (ATM) network and the routing protocol for intermittent advertisement of local state information throughout the network is the PNNI protocol.
50. The method according to Claim 47, wherein the communications network is an Internet Protocol (IP) network and the routing protocol for intermittent advertisement of local state information throughout the network is the Open Shortest Path First (OSPF) protocol.
51. The method according to Claim 49, wherein the notification message is a PNNI Database Summary packet in which said flag is provisioned.

52. The method according to Claim 51, wherein the topology state information is exchanged by bundling the topology state information into PNNI Topology State Elements (PTSE).

53. The method according to Claim 52, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP).

54. A network element for synchronization of topology state information between two network nodes in a communications network, the communications network comprising a routing protocol for intermittent advertisement of local state information throughout the network, the two network nodes comprising a requesting node which initiates a request for topology state synchronization and a replying node which receives said request and which communicates with the requesting node to provide topology state information to the requesting node which is not possessed by the requesting node when same initiates said request, the network element selectively operating in one of two modes of synchronization, wherein a first mode thereof effects topology state synchronization between the requesting node and the replying node which entails withdrawal, by the said requesting node and by the said replying node, of the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node, and wherein a second mode thereof effects topology state synchronization between the requesting node and the replying node which maintains the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node.

55. The network element according to Claim 54, wherein topology state information is exchanged between the requesting node and the replying node once the request for topology state synchronization has been made by the requesting node to the replying node.

56. The network element according to Claim 55, wherein prior to the said exchange of topology state information, the requesting node notifies the replying node that the said exchange is to take place according to one of the said first mode and the said second mode of synchronization.

57. The network element according to Claim 56, wherein said notification takes place by way of a flag in a notification message sent by the requesting node to the replying node.

58. The network element according to Claim 56, wherein the topology state information which is exchanged between the requesting node and the replying node is extracted from topology state databases each respectively associated with the said requesting and replying nodes.

59. The network element according to Claim 57, wherein the communications network is an Asynchronous Transfer Mode (ATM) network and the routing protocol for intermittent advertisement of local state information throughout the network is the PNNI protocol.

60. The network element according to Claim 57, wherein the communications network is an Internet Protocol (IP) network and the routing protocol for intermittent advertisement of local state information throughout the network is the Open Shortest Path First (OSPF) protocol.

61. The network element according to Claim 59, wherein the notification message is a PNNI Database Summary packet in which said flag is provisioned.

62. The network element according to Claim 61, wherein the topology state information is exchanged by bundling the topology state information into PNNI Topology State Elements (PTSE).

63. The network element according to Claim 62, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP).

64. A method for synchronization of topology state information between a first network node and a second network node in a communications network, the communications network comprising a routing protocol for exchange of local state information throughout the network, the first network node initiating a request for topology state synchronization and the second network node receiving said request and communicating with the first network node to provide topology state information to the first network node, the topology state synchronization taking place according to a first mode thereof wherein the said exchange of local state information, as it pertains respectively to the first network node and to the second network node, is not withdrawn.

65. The method according to Claim 64, wherein a second mode of topology state synchronization is provided, the second mode of topology state synchronization entailing withdrawal, by the first network node and by the second network node, of the said exchange of local state information as it pertains respectively to the first network node and to the second network node.

66. The method according to Claim 65, further comprising the step of selecting, prior to said request being initiated by the first network node to the second network node, between the first mode of topology state synchronization and the second mode of topology state synchronization.

67. The method according to Claim 66, wherein the second network node communicates with the first network node to provide topology state information to the first network node which is not possessed by the first network node when same initiates said request.

68. The method according to Claim 67, wherein prior to the said exchange of topology state information, the first network node notifies the second network node that the said exchange of topology state information is to take place according to one of the said first mode and the said second mode of synchronization.

69. The method according to Claim 68, wherein said notification takes place by way of a flag in a notification message sent by the first network node to the second network node.

70. The method according to Claim 69, wherein the topology state information which is exchanged between the first network node and the second network node is extracted from topology state databases each respectively associated with the said first and second network nodes.

71. The method according to Claim 70, wherein the communications network is an Asynchronous Transfer Mode (ATM) network and the routing protocol for exchange of local state information throughout the network is the PNNI protocol.

72. The method according to Claim 70, wherein the communications network is an Internet Protocol (IP) network and the routing protocol for exchange of local state information throughout the network is the Open Shortest Path First (OSPF) protocol.

73. The method according to Claim 71, wherein the notification message is a PNNI Database Summary packet in which said flag is provisioned.

74. The method according to Claim 73, wherein the topology state information is exchanged by bundling the topology state information into PNNI Topology State Elements (PTSE).

75. The method according to Claim 74, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP).

76. A network element for synchronization of topology state information between a first network node and a second network node in a communications network, the communications network comprising a routing protocol for exchange of local state information throughout the network, the first network node initiating a request for topology state synchronization and the second network node receiving said request and communicating with the first network node to provide topology state information to the first network node, the topology state synchronization taking place according to a first mode thereof wherein the said exchange of local state information, as it pertains respectively to the first network node and to the second network node, is not withdrawn.

77. The network element according to Claim 76, wherein a second mode of topology state synchronization is provided, the second mode of topology state synchronization entailing withdrawal, by the first network node and by the second network node, of the said exchange of local state information as it pertains respectively to the first network node and to the second network node, the network element selectively operating in one of the two said modes of topology state synchronization.

78. The network element according to Claim 77, wherein prior to said request being initiated by the first network node to the second network node, the network element selects between the first mode of topology state synchronization and the second mode of topology state synchronization.

79. The network element according to Claim 78, wherein prior to said request being initiated by the first network node to the second network node, the first network node notifies the second network node that the said exchange of topology state information is to take place according to one of the first mode of topology state synchronization and the second mode of topology state synchronization.

80. The network element according to Claim 79, wherein the second network node communicates with the first network node to provide topology state information to the first network node which is not possessed by the first network node when same initiates said request.

81. The method according to Claim 80, wherein said notification takes place by way of a flag in a notification message sent by the first network node to the second network node.

82. The method according to Claim 81, wherein the topology state information which is exchanged between the first network node and the second network node is extracted from topology state databases each respectively associated with the said first and second network nodes.

83. The method according to Claim 82, wherein the communications network is an Asynchronous Transfer Mode (ATM) network and the routing protocol for exchange of local state information throughout the network is the PNNI protocol.

84. The method according to Claim 82, wherein the communications network is an Internet Protocol (IP) network and the routing protocol for exchange of local state information throughout the network is the Open Shortest Path First (OSPF) protocol.

85. The method according to Claim 83, wherein the notification message is a PNNI Database Summary packet in which said flag is provisioned.

86. The method according to Claim 85, wherein the topology state information is exchanged by bundling the topology state information into PNNI Topology State Elements (PTSE).

87. The method according to Claim 86, wherein each PTSE is encapsulated within PNNI Topology State Packets (PTSP).

3. Detailed Description of Invention

FIELD OF THE INVENTION

The present invention relates generally to the field of network topology database re-synchronization in communications networks having topology state routing protocols and more particularly, to a method and apparatus for effecting network topology database re-synchronization in such networks. For example, the present invention is well-suited to database re-synchronization in the context of redundancy recovery following a nodal failure or a reset affecting an active routing entity associated with a network node.

BACKGROUND OF THE INVENTION

Topology state routing protocols are employed in communications networks in order to disseminate or advertise topology state information among nodes and node clusters within such networks. The advertised topology state information is in turn utilized to compute optimized paths for communications throughout a given network. As used in the present application, reference to topology state information signifies state information for the network domain as a whole. In certain network protocols, topology state information includes both link state information and nodal state information. For instance, link state information will include such attributes as link characteristics, link operational status, port identifiers and remote neighbour information concerning adjacent neighbour nodes. Nodal state information will include such attributes as node identifiers, peer group identifiers, distinguished node election status, distinguished node leadership status and local reachable address information.

Whereas topology state information will refer to state information for a network domain as a whole, the present application will make reference to local state information when dealing with state information which is locally originated by a particular network

node. Local link status information will reflect a given node's understanding of the status of communication with its peer nodes. Thus, local link status information, similarly to topology link status information, will also include such attributes as link characteristics, link operational status, port identifiers and remote neighbour information concerning adjacent neighbour nodes, but these will pertain to a given network node as opposed to a variety of nodes forming part of a network domain. Likewise, local nodal state information will comprise such attributes as node identifiers, peer group identifiers, distinguished node election status, distinguished node leadership status and local reachable address information. Again, these will pertain to a given node when reference is made to local nodal state information, instead of pertaining to the network domain as a whole when reference is made to topology nodal state information. In the present application, reference to state information will signify both topology state information and local state information.

In some known topology state protocols, certain nodes in a communications network may take on distinguished or additional responsibilities in order to make the routing function for the network operate properly. For instance, in the Open Shortest Path First (OSPF) IP routing protocol as described in J. Moy: "OSPF Version 2", STD 54, RFC 2328, dated April 1998, a node identified as the Designated Router (DR) would assume such responsibilities. Similarly, in the Private Network-Node Interface or Private Network-to-Network Interface (PNNI) protocol, responsibilities of this nature are assumed by a node termed the Peer Group Leader (PGL). The PNNI protocol is specified in the documents entitled: (i) "Private Network Interface Specification Version 1.0", ATM Forum document no. af-pnni-0055.000 dated March 1996, (ii) "Private Network-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB)", ATM Forum document no. af-pnni-0066.000 dated September 1996 and (iii) "Addendum to PNNI V1.0 for ABR parameter negotiation", ATM Forum document no. af-pnni-0075.000 dated January 1997, together with amendments found in (iv) "PNNI V1.0 Errata and PICS, ATM Forum document no. af-pnni-0081.000 dated May 1997 (hereafter all of the

foregoing documents (i) through (iv), inclusively, are collectively referred to as the "PNNI Specification"). The PNNI Specification is hereby incorporated by reference.

A given physical node within a network space may acquire distinguished network responsibilities of the type mentioned above by a process known as distributed election. In a scheme of distributed election, all nodes at a particular level of a network hierarchy will communicate to select the node which is to assume additional tasks or responsibilities in relation to the topology state protocol. Those skilled in this art will understand that performing the process of distributed election will take varying amounts of time depending on the particular network environment. As well, if due to downtime the distinguishing position is not being filled by a given network node, the routing functions of a portion of the network or of the network domain as a whole may exhibit reduced capabilities or inefficiency during the downtime interval. Thus, it can be expected that in communications networks which utilize topology state protocols, a recovery interval must be tolerated by the network routing system subsequent to the failure of a network node. For instance, this may occur to varying degrees of severity whenever the failed node impacts the functions of an elected network node having the additional responsibilities referred to earlier.

Certain routing protocols specify a given level of node redundancy. This redundancy is intended to reduce the recovery time of the network routing system in the event of a failure that affects a node which performs distinguished protocol functions of the kind mentioned previously. For example, in the OSPF protocol, the use of a Backup Designated Router (BDR) is specified. The Backup Designated Router is mandated to detect a failure affecting the currently appointed Designated Router. Upon detecting such a failure, the Backup Designated Router will be called upon to take recovery action to declare itself the new Designated Router in the place of the failed former Designated Router. All other routers on the affected portion of the shared network will thereafter be notified of the existence of the new Designated Router node. Thus, although it is not

necessary to re-execute a dynamic election process under the OSPF protocol following a failure which impacts a Designated Router node, a network routing outage of some duration will nevertheless be experienced by all routers and hosts on the shared network that were originally served by the failed Designated Router node. This is because the affected routers and hosts participate in recovering the functions of the network routing system following a failure which impacts their associated Designated Router node.

On the other hand, in the PNNI protocol, no provision is currently made for distinguished node redundancy. As such, the distributed election process and its associated protocol actions must be re-executed upon any failure affecting a distinguished network node. In the PNNI protocol, a physical node which performs the Peer Group Leader function at one level of the topology hierarchy may be performing this function at several other levels of the hierarchy. Thus, a failure affecting such a physical node may very well impact a large part of the aggregated network. Furthermore, there is no provision in the current PNNI protocol for a backup Peer Group Leader. Thus, a failure which affects a multilevel Peer Group Leader of the kind described above must be detected by all logical nodes which form part of the various Peer Groups that are represented by the multilevel Peer Group Leader. These logical nodes at different levels of the network hierarchy must thereafter elect a new Peer Group Leader. As with the example given previously in relation to the OSPF protocol, the failure of the Peer Group Leader may be known to many nodes and hence such nodes must generally all participate in recovering the affected functions of the routing system. Given this, the failure of a Peer Group Leader in a PNNI network may conceivably impact a large portion of the network and may in many circumstances cause disruption of the routing behaviour of the network for a period of time which may be unacceptable to service providers or end users.

The discussion above has addressed the impact of a failure affecting a network node which has distinguished responsibilities. However, it will be appreciated by those versed in this art that a failure concerning an ordinary physical or logical node which does

not possess distinguished responsibilities will also result in some measure of disruption to the routing capabilities of the neighbouring nodes or devices that are serviced by the failed ordinary node. Although in some node architectures it may be possible to retain certain network functions such as packet forwarding or call processing in the event of a routing function failure, topology state protocols such as OSPF and PNNI require each network node of a domain to synchronize a topology database with its neighbours before being admitted to the routing system. Such topology database synchronization must take place in these network protocols in order to recover from the failure of a node. The synchronization process may consume seconds or minutes in the overall scheme of recovery, depending on the circumstances. During the synchronization, network devices serviced by the failed node will be impacted and hence routing functionality may very well suffer disruption. While the discussion above has focussed on the challenges surrounding recovery from a nodal failure, those skilled in this art will understand that analogous problems arise stemming from other events which would require a node to undertake a synchronization of its topology database, for instance a reset of the routing processor associated with a network node.

Certain mechanisms have been developed in the prior art to ensure a switchover between distinct routers in a manner that is transparent to hosts which use a failed router. The Hot Standby Router Protocol described in T. Li, B. Cole, P. Morton and D. Li: "Cisco Hot Standby Router Protocol (HSRP)", RFC 2281, dated March 1998, and the IP Standby Protocol according to P. Higginson and M. Shand: "Development of Router Clusters to Provide Fast Failover in IP Networks", 9 Digital Technical Journal, No. 3, dated Winter 1997, are two examples of such transparent router switchover schemes. However, as will be explained in greater detail below, switchover mechanisms of this type do not generally ensure that the switchover will be universally transparent to the routers or nodes in the network beyond the particular hosts or nodes immediately adjacent the failed node. In the prior art, the failure of a node is typically recovered by means of a distinct and different node. It would therefore be advantageous to provide a mechanism

that would allow the failure of a routing component of a node to be recovered by another routing component of the same node in a manner transparent to all nodes but its immediate neighbours.

Accordingly, prior art topology state routing protocols present problems and challenges when faced with a situation of recovery from a nodal failure or with other situations which may require a node to synchronize its topology database once it has previously done so, and these problems and challenges arise whether or not the node immediately affected by the failure has distinguished responsibilities. First, known recovery mechanisms typically disrupt the routing functions of at least a part of a network and cause a service impact to certain of the devices utilizing the network. The portion of the network affected will vary in the circumstances. For instance, the impacted portion of the network can be expected to be more extensive for a node performing distinguished functions than is the case for a node that does not perform such functions. As well, the impacted portion can be expected to be more expansive for a failure concerning a PNNI Peer Group Leader than for one which influences an OSPF Designated Router. Second, the time required to recover from a node or link failure will vary, but may be in the order of up to several minutes or longer. As mentioned above, this time frame may be unacceptable to certain service providers or end users. Third, since many nodes will have to be made aware of the failure and are therefore required to participate in the recovery process, network resources in the nature of bandwidth and processing time will be diverted. This will detract from other network activities in general and may decrease the performance and stability of the network routing system in particular.

It is therefore generally an object of the present invention to seek to provide a method and apparatus for database re-synchronization in a network having a topology state routing protocol, particularly well-suited to the context of redundancy recovery following a nodal failure associated with the routing entity of a network node, and

pursuant to which some of the problems exhibited by alternative prior art techniques and devices may in some instances be alleviated or overcome.

SUMMARY OF THE INVENTION

According to a first broad aspect of the present invention, there is provided a method for recovery from a failure which affects an active routing entity in a communications network, the active routing entity being associated with a network node of the communications network, the communications network comprising a routing protocol for intermittent advertisement of local state information throughout the network and further comprising an inactive routing entity to which network connections of the network node can be diverted from the active routing entity upon the failure, the method comprising the steps of: (a) upon the failure, executing an activity switch between the active routing entity and the inactive routing entity, wherein network connections of the network node are diverted from the active routing entity to the inactive routing entity to thereby transform the inactive routing entity into a newly active routing entity; (b) following the activity switch, exchanging topology state information between the newly active routing entity and each immediately adjacent neighbour node of the network node associated with said failure such that the newly active routing entity and every said immediately adjacent neighbour node respectively possess synchronized topology state information; and wherein the exchange of topology state information between the newly active routing entity and each said immediately adjacent neighbour node is performed without withdrawal, by the network node associated with said failure and by each said immediately adjacent neighbour node, of the said intermittent advertisement of local state information as it pertains respectively to the network node associated with said failure and to each said immediately adjacent neighbour node.

According to a second broad aspect of the present invention, there is provided a network element for recovery from a failure in a communications network which includes a routing protocol for intermittent advertisement of local state information throughout the

network, the network element comprising: an active routing entity, wherein the active routing entity is associated with topology state information concerning the communications network; an inactive routing entity, wherein an activity switch is executed between the active routing entity and the inactive routing entity upon failure of the active routing entity to thereby divert network connections from the active routing entity to the inactive routing entity and transform the inactive routing entity into a newly active routing entity; a database synchronization processor, wherein the database synchronization processor effects an exchange of topology state information between the newly active routing entity and each immediately adjacent neighbour node of the network element following the activity switch such that the newly active routing entity and every said immediately adjacent neighbour node respectively possess synchronized topology state information, and wherein the said exchange of topology state information between the newly active routing entity and each said immediately adjacent neighbour node is performed without withdrawal, by the network node associated with said failure and by each said immediately adjacent neighbour node, of the said intermittent advertisement of local state information as it pertains respectively to the network element and to each immediately adjacent neighbour node.

According to a third broad aspect of the present invention, there is provided a method for synchronization of topology state information between two network nodes in a communications network, the communications network comprising a routing protocol for intermittent advertisement of local state information throughout the network, the two network nodes comprising a requesting node which initiates a request for topology state synchronization and a replying node which receives said request and which communicates with the requesting node to provide topology state information to the requesting node which is not possessed by the requesting node when same initiates said request, the method comprising the step of selecting, prior to said request being made by the requesting node to the replying node, between a first and a second mode of synchronization, the first said mode providing for topology state synchronization which entails withdrawal, by the said requesting node and by the said replying node, of the said

intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node, and the second said mode providing for topology state synchronization which maintains the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node.

According to a fourth broad aspect of the present invention, there is provided a network element for synchronization of topology state information between two network nodes in a communications network, the communications network comprising a routing protocol for intermittent advertisement of local state information throughout the network, the two network nodes comprising a requesting node which initiates a request for topology state synchronization and a replying node which receives said request and which communicates with the requesting node to provide topology state information to the requesting node which is not possessed by the requesting node when same initiates said request, the network element selectively operating in one of two modes of synchronization, wherein a first mode thereof effects topology state synchronization between the requesting node and the replying node which entails withdrawal, by the said requesting node and by the said replying node, of the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node, and wherein a second mode thereof effects topology state synchronization between the requesting node and the replying node which maintains the said intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node.

According to a fifth broad aspect of the present invention, there is provided a method for synchronization of topology state information between a first network node and a second network node in a communications network, the communications network comprising a routing protocol for exchange of local state information throughout the network, the first network node initiating a request for topology state synchronization and the second network node receiving said request and communicating with the first network

node to provide topology state information to the first network node, the topology state synchronization taking place according to a first mode thereof wherein the said exchange of local state information, as it pertains respectively to the first network node and to the second network node, is not withdrawn.

According to a sixth broad aspect of the present invention, there is provided a network element for synchronization of topology state information between a first network node and a second network node in a communications network, the communications network comprising a routing protocol for exchange of local state information throughout the network, the first network node initiating a request for topology state synchronization and the second network node receiving said request and communicating with the first network node to provide topology state information to the first network node, the topology state synchronization taking place according to a first mode thereof wherein the said exchange of local state information, as it pertains respectively to the first network node and to the second network node, is not withdrawn.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Redundancy techniques for network components or devices, such as hot redundancy techniques, are generally well known to those skilled in this art. With reference to Figure 1, these techniques will be explained using the illustrative example of a communications network in the form of a PNNI network domain 30. However, those skilled in this art will understand that the present invention may be applied or adapted to other types of networks as well, for instance Internet Protocol (IP) networks for which intermittent advertisement of local state information is accomplished by the Open Shortest Path First (OSPF) routing protocol. As well, the present invention is suited not only to situations of recovery from failures associated with a routing entity of a network node, but also to other contexts where it may be necessary or desirable for a network node to re-synchronize its topology database.

Topology State Routing Protocols and Topology Database Synchronization

The communications network 2 has a network domain 30 which is comprised of a plurality of network nodes 32 to 41, each of which typically are switching systems. The network nodes 32 to 41 are interconnected by way of physical or logical links 42 to 53 that respectively attach two given switching systems of the network domain. The network element or node 56 (also labeled "A.1.2") of the PNNI network domain 30 is shown as having assumed the role of Peer Group Leader for the parent Peer Group

labelled "PG(A)", and the presence of node 36 at the level of the parent Peer Group is the consequence of the leader status of the node 56. Node 36 also represents a network domain in the form of the child Peer Group 55 (also labelled "PG(A.1)") which comprises lower-level network nodes 56 to 60. The lower-level network nodes 56 to 60 are interconnected by way of physical or logical links 62 to 67 each attaching two given lower-level switching systems. The functions which define the Peer Group Leader of PG(A.1) are implemented on the switching system which contains lower-level node 56 (also labelled "A.1.2"). PG(A.1) is a child peer group of PG(A) and is represented in PG(A) as a logical node 36, implemented within the physical switching system 56. Similarly, the parent Peer Group labelled "PG(A)" may itself be a child Peer Group represented at a higher level of the routing hierarchy by a single logical node (not shown).

According to known redundancy techniques, the particular node, switch or other network entity for which fault tolerance protection is desired usually provides at least two routing processors within a single network element. A routing processor performs the function of maintaining connectivity to its adjacent neighbour nodes and of sharing topology state information with those nodes. Preferably, the routing processors will be configured by way of distinct physical components. For instance, the physical components may each be in the form of distinct hardware cards provisioned within the same network switch, such as within the network node 56 ("A.1.2"). Where two processors are provided for redundancy purposes, one of the physical components in question will assume the role of the active routing entity for the redundant network element and the other of the physical components will thus be an inactive routing entity therefor.

Upon detecting the failure of the active routing entity, the inactive routing entity is called into service to take over the functions of the failed active routing entity. This procedure is termed an activity switch. Because both of these routing entities are associated with the same node (e.g. the network node 56), the node itself need not

relinquish any of its distinguished responsibilities. As well, only immediate neighbouring nodes of the failed node in the form of immediately adjacent parent Peer Group nodes (e.g. the network nodes 34, 35, 37, 38) and any immediately adjacent child Peer Group nodes (e.g. the network nodes 57, 59, 60) need be called upon or otherwise enlisted to take part in network recovery. However, as discussed below, current topology state protocols may nevertheless cause more nodes than those immediately neighbouring the failed node to be impacted during the recovery process (e.g. the network nodes 32, 33, 39, 40, 41, 58), thereby increasing the time required for recovery to take place as well as the network resources consumed in the process.

Existing capabilities and techniques may be utilized to implement a scheme of redundancy protection in a given network architecture, such as the PNNI network domain 30. For instance, these capabilities and techniques may include the management of activity status within the various network nodes and the synchronization of state information between the active and inactive routing components. This state information for a network topology is typically stored in a synchronization database, also called a topology database, which is associated with each network node of a routing domain. Typically, the synchronization database will be stored within the network nodes in question. Database synchronization is an existing topology state routing protocol mechanism which ensures that adjacent nodes within a network share a common view of the overall topology of the network. Some signalling protocols, for instance ITU-T Q.2931, have mechanisms such as Status Enquiry schemes to perform a synchronization of the call state between two network nodes.

One problem with some known schemes of redundancy is that when a failure occurs at a network node such as node 56, which implements a higher-level node 36, the affected links in PG (A) 44, 45, 47 and 51 to and from the failed node stop being advertised after some time or while a new PGL in PG (A.1) begins taking over responsibility to implement higher level node 36. In other words, when the newly active

routing processor initiates a database synchronization with its peers, the current PNNI protocol will call for the advertisement of local state information from each of the nodes involved in the synchronization to be removed or withdrawn until such time as the synchronization has taken place. Thus, the failed node will stop advertising its local state information and the nodes which neighbour the failed node will likewise stop advertising their respective local state information. This is true of any synchronization which is performed under the currently known protocol, where synchronization is required after an activity switch or a processor reset situation. In the existing PNNI protocol, a failed node that provides hot redundancy capability will be required to synchronize its topology database with neighbouring nodes. This is because the active and inactive routing processors of such a node will generally have exchanged topology state information internally prior to the failure, and this exchange will have taken place periodically. As such, there may very well be a loss of topology status information in the interval between the latest exchange of state information and the failure, which loss of information is experienced by the inactive routing processor upon the activity switch from the formerly active and failed routing processor.

When a failed node such as the node 56, which also implements higher level node 36 and which is provided with redundancy capability according to the prior art, is restarted by way of an activity switch, the newly active routing entity resulting from the activity switch must therefore re-establish the affected links 62, 63 and 64 to and from node 56 and the affected links 44, 45, 47 and 51 to and from the higher level node 36. Where a failed node such as the node 56 is a Peer Group Leader for a given number of child peer group nodes 56, 57, 58, 59 and 60, other links in the child peer group must also be re-established. In order to re-establish their existence as part of the network topology, a failed node and its neighbours must first learn of each other's presence. In the PNNI protocol, two-way Hello packets will accomplish this function of neighbour discovery in the case of lowest level peers connected by physical links or VPCs. Next, the failed node and each of its neighbours will exchange topology database summary information with

one another. Typically in the PNNI protocol, the newly reacquainted nodes will first exchange PNNI Topology State Element (PTSE) headers to determine whether or not the nodes are in synchronization. If they are not, synchronization will take place. When a node receives PTSE header information that advertises a PTSE it does not yet have, it requests the advertised PTSE and updates its topology database with that requested PTSE once it is received.

Once synchronization is completed, local state information is advertised among the failed node and its neighbour nodes within the network topology of network domain 30, whereby each node will advertise its reachability to all other neighbour nodes. For example, in the PNNI protocol, this advertisement takes place by means of a regular process known as flooding. As previously explained, this known process of re-establishment adversely impacts upon the routing functions of the network. Moreover, network resources are consumed during re-establishment of the failed node 56 according to known techniques.

Modifications to the Known Neighbouring Peer Finite State Machine, Neighbouring Peer Data Structure and Database Summary Packet Structure

According to an illustrative embodiment, the invention may be adopted in the context of the PNNI protocol by making various modifications to the existing protocol of Figure 2. As described in greater detail in what follows, these modifications generally relate to the known neighbouring peer Finite State Machine (FSM) 5 and its associated transition events, to other aspects of the neighbouring peer data structure and to the Database Summary packet structure. With reference to Figure 3, two additional states for effecting database re-synchronization are defined for the modified neighbouring peer Finite State Machine 10. The additional states are termed the Exchange in Full State 22 and the Loading in Full State 24. The Exchange in Full State 22, the Loading in Full State 24 and their associated state transitions are described in greater detail below.

New transition events causing a neighbour peer state change are also added to the existing state transitions found at Section 5.7.3 of the PNNI Specification. These new state transitions may be termed the DS (Resynch) Mismatch event and the (Re)Synch Done 25 event. Other additional state transitions such as Negotiation (in Full) Done 28, Loading (in Full) Done 27 and Exchange (in Full) Done 26 are employed with the new states of the present invention. These additional state transitions respectively mirror the existing state transitions of Negotiation Done 15, Loading Done 21 and Exchange Done 17. This is explained in greater detail below with reference to Table 1. With the exception of the added states and state transitions which relate to them, the various states comprising the Finite State Machine 10 are otherwise as ordinarily defined in the PNNI Specification at Section 5.7.2 thereof.

The neighbouring peer data structure, which is normally present in the PNNI protocol and is defined at Section 5.7.1 of the PNNI Specification, is also modified according to the invention by the definition of an additional timer and associated timer interval when compared to the existing protocol. The additional timer and interval may be termed respectively the Resynch Inactivity Timer and the Resynch Inactivity Interval for convenience of description. When this new timed interval expires, it will signal a potential problem with a database re-synchronization. In such a case, the node re-synchronization will be required to perform a database synchronization from the Negotiating State 14, as would have normally occurred if the database synchronization had originally taken place in the existing PNNI protocol.

Lastly, an added bit is provided according to one implementation of the present invention for the Database Summary packet structure. The additional bit is described more fully below and may be termed a Synch in Full (SF) Bit. This added bit to the Database Summary packet structure is intended to signal to adjacent nodes that a re-

synchronization is taking place, for example due to a failure of a routing entity, such that this re-synchronization does not ensue according to the existing PNNI protocol.

With reference to Figure 2, the known neighbouring peer Finite State Machine 5 is used in the existing PNNI Specification to describe the status of database synchronization and flooding which is ongoing between a given node and each of its adjacent neighbouring peers. As known, the specified Finite State Machine 5 for neighbouring peers has an Initial State 12 (labelled "NPDown") which would indicate that there are no active links to the particular neighbouring peer. The first step to creating an adjacency with a neighbouring peer is the Negotiating State 14, known to those skilled in this art. The triggering event to transition to the Negotiating State 14 from the Initial State 12 is termed an Add Port event 13. Here, a Hello state machine for a link to the neighbouring peer has typically reached its 2-WayInside state, as known to those conversant with this art. In the Negotiating State 14, two neighbouring peers will decide which of the nodes is to be the Master for the purpose of database exchange, and an initial DS sequence number is selected. Those skilled in this art will understand that the DS sequence number is used to identify individual Database Summary packets.

Once the Negotiating State 14 is completed, the neighbouring peer nodes will receive Database Summary packets from the node in question during the Exchanging State 16. Thus, the Negotiation Done event 15 will transition the Finite State Machine 10 to the Exchanging State 16. As already known in the PNNI protocol, in the Exchanging State 16 the node in question describes its topology database to the neighbouring node. Once the neighbouring node has processed the Database Summary packets, it can then proceed to request its required PTSE information. If such information is required, the Exchange Done event 17 will transition the Finite State Machine 5 to the Loading State 18.

During the Loading State 18, required PTSE information is requested by the neighbouring node and at least one PTSE header has not yet been received by that node. Lastly in the known PNNI protocol, the Full State 20 is achieved upon either of two events occurring. First, the Full State 20 is arrived at once the receipt of PTSE information is completed following the Loading State 18 by way of the Loading Done event 21. Alternatively, the Full State 20 may be reached directly following the Exchanging State 16 by way of the Synch Done event 19, if the processing of Database Summary packets reveals that no PTSE information is required by the neighbouring node. In the Full State 20, all PTSE information known to be available from the neighbouring peer is possessed by the node in question and links to the neighbouring peer can thereafter be advertised by way of PTSEs.

As mentioned above and with particular reference to Figure 3, new states are added by way of one embodiment of the present invention to the known Finite State Machine 5 to achieve the modified Finite State Machine 10. These include the Exchange in Full State 22 and the Loading In Full State 24. Where a node in the Full State 20 has already synchronized its topology database prior to a failure or prior to some other requirement for further synchronization, and the node thereafter requires a database re-synchronization, it sends initialized Database Summary packets to its neighbouring peer nodes. As alluded to above, these Database Summary packets will have their respective Sync in Full (SF) Bits set. The Negotiation (in Full) Done event 28 then transitions the Finite State Machine 10 to the Exchange in Full State 22. In the Exchange In Full State 22, a node will attempt to describe its topology database to its neighbour nodes without withdrawing its advertisements of local state information to the neighbours that were advertised when reaching the Full State 20. Similarly, the neighbour nodes with which the node in question is synchronizing will also not withdraw advertisement of their respective local state information to the node requesting a database re-synchronization. Database Summary packets will be sent to the neighbouring peers of a node requesting resynchronization. These Database Summary packets will also have their respective Sync

in Full (SF) Bits set. If, as a result of processing Database Summary packets, the failed node determines that no PTSE packets are required, then the Finite State Machine 10 transitions to the Full State 20 by means of the (Re)Sync Done event 25.

On the other hand, if PTSE packets are required, the Finite State Machine 10 executes the Exchange (in Full) Done event 26 in order to transition to the Loading in Full State 24. The required PTSEs will then be requested by the node requiring re-synchronization with links to the neighbouring peer nodes being still advertised as in the Full State 20. Once all requested PTSE packets have been received by the node requesting resynchronization, the Loading (in Full) Done event 27 will transition the Finite State Machine 10 to the Full State 20. Thus, the node requesting resynchronization will send and receive Database Summary, PTSE Request and PTSP packets while its peer data structure is in the Full State 20, Exchange in Full State 22 and Loading in Full State 24. The same state transitions described above will also occur in relation to the neighbouring peer nodes of the failed node which is requesting synchronization.

As such, according to the present invention, two database synchronization procedures are provided, one which operates when the peer data structure is in the Full State 20 and the other which operates when the peer data structure is in the Negotiating State 14. Links between lowest-level neighbouring peers may only be advertised in PTSEs when the neighbouring peer Finite State Machine 10 is in the Full State 20, Exchange in Full State 22 or Loading in Full State 24. Thus, for neighbouring lowest-level peers connected by physical links or VPCs, changes into the Full State 20 from states other than the Exchange in Full State 22 or Loading in Full State 24 and changes out of the grouping of states consisting of the Full State 20, Exchange in Full State 22 and Loading in Full State 24 will cause new instances of one or more PTSEs for a synchronizing node or for a resynchronizing node to be originated or flushed.

In addition to the state transitions which are illustrated in Figure 2, the known PNNI protocol has four additional events. These are termed the DSMismatch, BadPTSERequest, DropPort and DropPortLast events. Each of the DSMismatch and BadPTSERequest events forces a state transition to the Negotiating State 14. The DSMismatch will occur whenever a Database Summary packet has been received with any one of the following occurring: (i) it has an unexpected DS sequence number, (ii) it has its Initialize Bit unexpectedly set, or (iii) it has an unexpected setting of its Master Bit. According to the present invention, the known DSMismatch event is modified so that the event will also occur whenever a Database Summary Packet has an unexpected setting of the Sync in Full Bit. As well, according to the present invention, a DSMismatch event will also occur when the DS Resynch Inactivity Timer, previously mentioned, expires. Any of the foregoing conditions indicates that an error has occurred in the database synchronization process. The known DropPort event causes no state change, and the known DropPortLast event forces the NPDown State 12. In the known DropPort event, a Hello state machine for a link to the neighbouring peer will have exited the 2-WayInside state. In the known DropPortLast event, it is determined that all ports to the neighbour node have been dropped.

As introduced above, an event which may be termed the DS (Resynch) Mismatch event is defined to take place whenever a Database Summary packet is received with its Synch in Full Bit set and such a packet has its Initialize Bit unexpectedly set. As was the case for the discussion of the DSMismatch event in the known PNNI protocol, the occurrence of the DS Resynch Mismatch event analogously signifies that an error has taken place in the process of database re-synchronization. The DS Resynch Mismatch will indicate that database re-synchronization will be re-attempted without the Finite State Machine 10 dropping to the Negotiating State 14.

The neighbouring peer Finite State Machine 10 according to one embodiment of the present invention will next be discussed in greater detail with reference to Table 1, set

out below. In Table 1, the new Exchange in Full State 22 and Loading in Full State 24 are displayed along with potential transition events for these states in summarized form, both existing in the known PNNI protocol and as newly added or modified by way of the present invention. Every cell in the table represents a pairing of a transition event and the indicated state as the current state of the node at the onset of the specified transition event. Each cell reflects the new state achieved as a result of the specified transition event together with an action to be taken by the node in question. Table 1 only displays those aspects of the existing neighbouring peer Finite State Machine 5 of the PNNI protocol which are modified by the present invention or which are supplemented by additional functions or procedures for implementing the present invention, so as to arrive at the modified Finite State Machine 10. These modifications and additions to the existing PNNI protocol are discussed in further detail below.

TABLE 1: Modifications to the Existing PNNI Neighbouring Peer Finite State Machine

Transition Events	States						
	NPDnwn	Negotiating	Exchanging	Loading	Full	Exchange in Full	Loading in Full
Add Port	Ds1 Negotiating	--	--	--	--	Ds8 Exchange in Full	Ds8 Loading in Full
Negotiation Done	--	--	--	--	--	FSM_ERR	FSM_ERR
Negotiation (in Full) Done	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds11 Exchange in Full	FSM_ERR	FSM_ERR
Exchange Done	--	--	--	--	--	FSM_ERR	FSM_ERR
Exchange (in Full) Done	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds3 Loading in Full	FSM_ERR
Sync Done	--	--	--	--	--	FSM_ERR	FSM_ERR
(Re)Synch Done	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds12 Full	FSM_ERR
Loading Done	--	--	--	--	--	FSM_ERR	FSM_ERR

Loading (in Full) Done	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds12 Full
DS Mismatch	--	--	Ds5 Negotiating	Ds5 Negotiating	--	Ds6 Negotiating	Ds6 Negotiating
DS (Resync) Mismatch	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	FSM_ERR	Ds13 Full	Ds13 Full
BadPTSE Request	--	--	Ds5 Negotiating	Ds5 Negotiating	--	Ds6 Negotiating	Ds6 Negotiating
DropPort	--	--	--	--	--	Ds9 Exchange in Full	Ds9 Loading in Full
DropPort Last	--	Ds10 NPDown	Ds10 NPDown	Ds10 NPDown	Ds10 NPDown	Ds10 NPDown	Ds10 NPDown

As in the existing PNNI Specification, at Section 5.7.4 thereof, FSM_ERR denotes an internal implementation error. Thus, the event Negotiation (in Full) Done 28 should not normally occur while the Finite State Machine 10 is any state except for the Full State 20. Likewise, the events Exchange (in Full) Done 26 and (Re)Sync Done 25 should not occur during any states other than the Exchange in Full State 22. As well, the Loading (in Full) Done event 27 is not expected to occur in any of the states except for the Loading in Full State 24. Moreover, the DS (Resync) Mismatch event is not expected to occur in any of the states except for the Exchange in Full State 22 and the Loading in Full State 24.

Each of the existing events Negotiation Done 15, Exchange Done 17, Sync Done 19 and Loading Done 21 are not provisioned to occur in the new states Exchange in Full 22 and Loading in Full 24 of the present invention. For this reason, the protocol error condition FSM_ERR is reflected under Table 1 for the mapping of the said existing events to the said new states. The existing events are replaced by the analogous events Negotiation (in Full) Done 28, Exchange (in Full) Done 26, (Re)Sync Done 25 and Loading (in Full) Done 27, as explained more fully below.

Where the event AddPort takes place while the node in question is in its NPDown State 12, the known procedures described in the PNNI protocol for the action termed Ds1

in Section 5.7.4 of the PNNI Specification are followed. Generally, the procedures call for the sending of Database Summary packets with no PTSE summaries. These Database Summary packets are retransmitted at the time interval specified by the known DS Rxmt Timer of the neighbouring peer data structure. The difference with the Ds1 procedures of the present invention is that the Database Summary packet will contain an additional bit in the form of the Sync in Full Bit previously mentioned. In the Ds1 procedures referred to in Table 1, the Sync in Full Bits of the Database Summary packets in question are not set. Where an AddPort event takes place during the Exchange in Full State 22 or during the Loading in Full State 24, the state machine will remain in the same state. In the case of lowest-level neighbouring peers connected by physical links or VPCs, the port ID is added to the Port ID list of the neighbouring peer data structure. As well, a link to the neighbouring peer will be added and a new instance of a PTSE will be originated. This set of actions is the same as specified in the PNNI protocol as it exists and is denoted as action Ds8 in Section 5.7.4 of the PNNI Specification.

If the Finite State Machine 10 is in its Full State 20 and the event Negotiation (in Full) Done 28 takes place, a transition to the Exchange in Full State 22 is triggered and the set of actions denoted as Ds11 in Table 1 will take place. Under this set of actions termed Ds11, and as is the case with the actions termed Ds2 in the known PNNI protocol, the node in question will begin sending a summary of the contents of its topology database to the neighbouring peer in the form of Database Summary packets. The procedures followed upon this transition to the Exchange in Full State 22 are identical to those denoted as Ds2 in Section 5.7.4 of the PNNI Specification, with the exception that the Database Summary packets sent by the node will have the Sync in Full Bit set. As well, according to the present invention, the Resynch Inactivity Timer will be started as part of the Ds2 procedures upon the state transition if this timer is not already running.

Where an Exchange (in Full) Done 26 event is injected during the Exchange in Full State 22 of the Finite State Machine 10, a transition to the Loading in Full State 24

will take place. The Ds3 action as described in Section 5.7.4 of the existing PNNI Specification thereafter ensues. Namely, the DS Rxmt Timer will be stopped if not previously stopped. PTSE Request packets will be sent or will continue to be sent to the neighbouring peer in question or to other neighbouring peers, as known to those in this art. Each of the PTSE Request packets will request some of the neighbouring peer's more recent PTSEs which were previously discovered but have not yet been received. These PTSEs will be listed in the PTSE Request list in the neighbouring peer data structure.

Whenever a (Re) Sync Done event 25 is injected while the state machine resides in its Exchange in Full State 22, the Finite State Machine 10 will transition to the Full State 20. This event associated with the Exchange in Full State 22 is termed the (Re) Sync Done event 25 to distinguish the event from its counterpart taking place during the Exchanging State 16.

Whenever a Loading (in Full) Done event 27 is injected while the Finite State Machine 10 resides in its Loading in Full State 24, the state machine will transition to the Full State 20. This event associated with the Loading in Full State 24 is termed the Loading (in Full) Done event in Figure 1 to distinguish the event from its counterpart taking place during the Loading State 18.

In the situation of a Loading (in Full) Done event 27 taking place during the Loading in Full State 24 or in the situation of a (Re) Sync Done event 25 taking place during the Exchange in Full State 22, the DS Rxmt timer and the Resynch Inactivity Timer introduced in the illustrative embodiment of the present invention and mentioned above will be stopped. These actions following the state changes in question are termed Ds12 in Table 1. This signifies that database re-synchronization is completed.

Where either a DSMismatch event or a BadPTSERequest event is injected during either of the Exchanging State 16 or the Loading State 18, a state transition to the Negotiating State 14 is effected. As well, the known procedures denoted Ds5 in the existing PNNI protocol are initiated following the state transition in question. These procedures are described in Section 5.7.4 of the PNNI Specification. According to the present invention, however, the Resynch Inactivity Timer will also be stopped if not previously stopped, as is the case with the Peer Delayed Ack Timer, DS Rxmt Timer and Request Rxmt Timer of the known protocol. According to the present invention, the Sync in Full Bit of the Database Summary packets sent by the node in the known Ds5 procedures will not be set.

The events DS Mismatch and BadPTSERequest of Table 1 which occur during either of the Exchange in Full State 22 or the Loading in Full State 24 will cause a transition to the Negotiating State 14. The actions to be taken will be as specified above in the case of the same event which occurs during either of the Exchanging or Loading states, with the exception that PTSEs will be modified to remove any advertising of links to the neighbour. The latter step is known in the existing PNNI protocol, and is described as action Ds6 in Section 5.7.4 of the PNNI Specification.

The Finite State Machine 10 is executed with the event DS Mismatch whenever the Resynch Inactivity Timer expires. This forces the node to withdraw all links to the non-responsive neighbour which are currently being advertised in PTSEs. The Finite State Machine 10 is then forced to the Negotiating State 14 where the node will once again attempt to synchronize databases in the known manner starting from the Negotiating State 14. In the event of a successful database re-synchronization, the Resynch Inactivity Timer is stopped.

Still with reference to Table 1, when a DS (Re Sync) Mismatch occurs during the Exchange in Full State 22 or the Loading in Full State 24, the Full State 20 is reverted to.

The action taken upon this transition event will result in the Peer Delayed Ack Timer, DS Rxmt Timer and Request Rxmt Timer being stopped if not previously stopped. All of the said timers are known in the existing PNNI protocol. The Peer Retransmission List, Peer Delayed Acks List, PTSE Request List and all related timers are also cleared. The exchange of database summaries with the Sync in Full Bit set must start over again. Then, the node in question increments the DS sequence number for this neighbouring peer, declares itself master by setting the Master Bit to a value of one, and will start sending Database Summary packets with the Initialize, More, Master and Sync in Full Bits set. No PTSE summaries are included with these packets. Lastly, the DS Rxmt Timer is started and the Database Summary packet is retransmitted each DSRxmtInterval time, if no DB Summary packet is received to acknowledge it. All of these actions upon the state change to the Full State 20 have been termed Ds13 in Table 1.

The DropPort event of Table 1 will result in the Finite State Machine 10 retaining the Exchange in Full State 22 and the Loading in Full State 24, wherever the Finite State Machine 10 is respectively in those same states. As with the Ds9 procedures of Section 5.7.4 of the existing PNNI protocol, the link in question is removed from the Port ID list in the corresponding neighbouring peer data structure. If there is a PTSE advertising that link, a new instance of the affected PTSE will be originated. If the link is the last active link to the neighbour, the DropPortLast event will be generated. As shown in Table 1, the DropPortLast event will transition the Finite State Machine 10 to the NPDown State 12 whenever the Finite State Machine 10 is in either of the Exchange in Full State 22 or the Loading in Full State 24. As known in the PNNI protocol and termed Ds10 in Section 5.7.4 of the PNNI Specification, the Peer Delayed Ack Timer, DS Rxmt Timer and Request Rxmt Timer are stopped if not previously stopped. As known in the existing Ds10 procedure, the Peer Transmission List, Peer Delayed Acks List and PTSE Request List are cleared together with their related timers. However, according to the present invention, the Resynch Inactivity Timer will also be stopped, if not previously stopped.

Sending Database Summary Packets

The sending of Database Summary packets is generally as specified by the known PNNI protocol, with the exceptions noted below. However, in the Full State 20, a node may initiate a database resynchronization by sending empty Database Summary packets with the Initialize, More, Master and Synch in Full Bits set. The Resynch Inactivity Timer is started when the first such Database Summary packet is sent, if this timer is not already running. When sending such Database Summary packets, the DS Rxmt Timer must be restarted. These packets are retransmitted by the node initiating a database resynchronization every DS Rxmt Interval seconds, when the DS Rxmt Timer fires.

A node in its Full State 20 will also send Database Summary packets in response to received Database Summary packets with the Initialize, More, Master, and Sync in Full Bits set from a neighbouring peer that is requesting a database re-synchronization. When the node initiates a database re-synchronization, it sends Database Summary packets with the Synch In Full, Initialize, More and Master Bits set to a value of one. A node that responds to a neighbour node requesting a database re-synchronization will set the Synch in Full, Initialize, More and Master Bits, as described more fully below. When a node responds to the first such Database Summary packet to a neighbour peer requesting a database re-synchronization, it starts the Resynch Inactivity Timer, if it is not already running.

In the Negotiating State 14, the node in question will send empty Database Summary packets as in the existing protocol, with the exception that the Sync in Full Bit will not be set in the Database Summary packet which has been modified according to the present invention to include an additional bit in the form of the said Sync in Full Bit. In the Exchange in Full State 22, the node in question will send Database Summary packets identical to those in the known Exchanging State 16, but the Sync in Full Bit will be set in such Database Summary packets.

Receiving Database Summary Packets

The processing of a received Database Summary packet by a node is next explained. The receiving of Database Summary Packets is generally as specified by the known PNNI protocol, with the exceptions noted below. If a Database Summary packet is accepted, a number of packet fields are saved in the corresponding neighbouring peer data structure in the known record identified as the Last Received Database Summary Packet's Identifying Information. Thus the packet flags consisting of the Initialize, More, Master and Reserved bits are saved in this manner, together with the DS sequence number, all as known in the existing PNNI protocol. However, the Sync in Full Bit of the present invention is also saved together with the foregoing bits. Where these saved items are set identically in two consecutive Database Summary packets received from a neighbouring peer, the second such Database Summary packet will be deemed a duplicate in the processing of received Database Summary packets.

As in the known PNNI protocol, a Database Summary Packet must be ignored if the packet is received when the Finite State Machine 10 is in the NPDown State 12.

In the existing PNNI Specification, a Database Summary Packet which is received when the Finite State Machine 5 is in the Negotiating State 14 and which matches one of the two cases specified in Section 5.7.6 of the existing PNNI Specification under the heading "Negotiating", will execute the Finite State Machine 5 with the event Negotiation Done and with a transition to the Exchanging State 16. In the first of these cases of the known protocol, the Initialize, More and Master Bits are set to a value of one. In the present invention, in addition to the Initialize, More and Master Bits having the foregoing values, the Sync in Full Bit must be set to a value of zero for a packet to be accepted as next in sequence and processed further in the manner already known according to the first case discussed under the heading "Negotiating" of Section 5.7.6. In the second of these cases of the known protocol, the Initialize and Master Bits are set to a value of zero. In the present invention, in addition to the Initialize and Master Bits having the foregoing

values, the Sync in Full Bit must be set to a value of zero for a packet to be accepted as next in sequence and processed further in the manner already known according to the second case discussed under the heading "Negotiating" of Section 5.7.6.

For a Database Summary packet which is received when the Finite State Machine 10 is in the Exchanging State 16, the procedures set out in the existing PNNI Specification under the heading "Exchanging" of Section 5.7.6 are followed, with one exception. Namely, a new condition is inserted immediately following the third existing step enumerated under the said heading of Section 5.7.6. This new condition requires that if the Sync in Full Bit is set, the event DS Mismatch will be generated and processing of the Database Summary Packet will be stopped. The procedures to be followed for receipt of a Database Summary packet in the Exchanging State 16 are otherwise as described in the existing PNNI Specification.

The receiving of Database Summary packets when the node is in its Loading State 18 signifies that the node has sent and received an entire sequence of Database Summary packets. Thus, the only packets so received should be duplicates. Any other Database Summary packets received must generate the event DS Mismatch, causing a reversion to the Negotiating State 14 with the two peers synchronizing their databases again according to the known PNNI protocol. The procedure followed when receiving a Database Summary packet in the Loading State 18 is therefore as specified in Section 5.7.6 of the existing PNNI protocol.

Where the node is in the Full State 20 and the received Database Summary packet has its Sync in Full Bit set, then if the received packet matches one of the two cases which follows, the Finite State Machine 10 must be executed with the Negotiation (in Full) Done event 28. The Finite State Machine 10 is thereby caused to transition to the Exchange in Full State 22. The packet is thereafter accepted as next in sequence and is processed further as set out below.

Case 1: Receiving Node is a Slave

This case covers the situation in which the Initialize, More and Master Bits are set to a value of one, the contents of the packet are empty, and the neighbouring peer Node ID is larger than the Node ID associated with the receiving node.

In this situation the node in question is a Slave, and upon generating the Negotiation (in Full) Done event 28, the slave node takes the following sequence of actions. First, the DS Rxmt Timer is stopped. Next, the Resynch Inactivity Timer is started if the timer is not already running. Then, the Master Bit is set to a value of zero (indicating the given node is a Slave), the Initialize Bit is also set to a value of zero, the Sync in Full Bit is set to a value of one, the DS sequence number is set to that specified by the Master node, and a Database Summary packet is sent to the Master including that portion of the database summary information for the node in question as set out in Section 5.7.5 of the existing PNNI specification.

Case 2: Receiving Node is a Master

This case covers the situation in which the Initialize and Master Bits are set to a value of zero, the DS sequence number of the packet is equal to the DS sequence number of the node in question (indicating an acknowledgement), and the Node ID of the neighbouring peer is smaller than that of the given node.

In this case the node in question is a Master node. Upon generating the Negotiation (in Full) Done event 28, the Master node must take the following sequence of actions. First, the DS Rxmt Timer must be stopped. Then, the contents of the received Database Summary packet must be acknowledged as

having been received and these contents are thereafter processed in the manner known in the existing PNNI protocol, as explained on pages 94 and 95 in Section 5.7.6 of the PNNI Specification under the heading "Exchanging State". The DS sequence number must be incremented by a value of one, the Sync in Full Bit is set to a value of one, and the Initialize Bit must be set to a value of zero. The Database Summary packet must be sent to the Slave node including that portion of the database summary information for the node in question as set out in Section 5.7.5 of the existing PNNI Specification, and the DS Rxmt Timer must be restarted. The Resynch Inactivity Timer is started if it is not already running.

If neither of Cases 1 or 2 above are applicable, the packet in question is verified as to whether it is a duplicate. If the packet is a duplicate, it is ignored. Otherwise, if the packet is not a duplicate then an error has occurred during re-synchronization and the Finite State Machine 10 must thereafter be executed with the event DS Mismatch.

If a node is in the Full State 20 and the received Database Summary packet does not have its Sync in Full Bit set, then the packet in question is expected to be a duplicate. Any other Database Summary packet with the Sync in Full Bit not being set will generate the event DS Mismatch, thereby causing the Finite State Machine 10 to revert to the Negotiating State 14 and the two neighbouring peers in question to re-synchronize their databases. According to the present invention, the procedures followed when receiving a Database Summary packet with the Sync in Full Bit not being set are the same as those followed in the known Exchanging State 16, except that packets accepted as the next in sequence must generate the event DS Mismatch and that further processing of such packets is stopped. The receipt of packets with an inconsistent Master Bit or with the Initialize Bit set to a value of one must also generate the event DS Mismatch.

Where the state of the node is the Exchange in Full State 22, a received Database Summary packet will be processed according to the present invention by executing the

following conditional steps. Where any one of these conditional steps has tested true, none of the remaining steps need be tested or executed.

- 1) If a Database Summary packet is received by the node acting as a Master, processing of the packet will be stopped if the packet is determined to be a duplicate. The determination of a duplicate packet was discussed previously.
- 2) If on the other hand the node is acting as a Slave, the response to the receipt of a duplicate packet is to retransmit the last Database Summary packet sent to the Master and then to stop processing the received Database Summary packet.
- 3) If the packet is not a duplicate in any event and the state of the Master Bit is inconsistent with the Master/Slave state of the connection, the event DS Mismatch is generated and the processing of the packet is stopped.
- 4) Where the Master Bit is consistent but the Sync in Full bit is not set, the event DS Mismatch will be generated and processing of the packet will be stopped.
- 5) Where all of the previously enumerated conditions have tested false and the Initialize Bit is set, the event DS Resynch Mismatch will be generated and processing of the packet will be arrested.
- 6) Where all of the previously enumerated conditions have tested false and the node is a Master, then the packet is accepted and further processed where the DS sequence number of the packet equals the DS sequence number for the node. This signifies that the packet is the next in sequence. The processing in this case is performed in the following manner.

The DS Rxmt Timer is stopped. Following this, the contents of the most recently received Database Summary packet are acknowledged as having been received and the contents thereof are then processed in the manner known in the existing PNNI protocol, as explained on pages 94 and 95 in Section 5.7.6 of the PNNI Specification under the heading "Exchanging". The DS sequence number is then incremented by a value of one. Next, if the node has already sent its entire sequence of Database Summary packets and the received packet has its More Bit set to a value of zero, the event (Re)Synch Done 25 is generated if the PTSE Request List is empty and the event Exchange (in Full) Done 26 is generated if the PTSE Request List is not empty. The determination of whether a node has sent its entire sequence of Database Summary packets is made if the previous Database Summary packet sent by the node also had its More Bit set to a value of zero. If the entire sequence of Database Summary packets is not yet received, a new Database Summary packet is sent to the Slave and the DS Rxmt Timer is restarted.

7) Where all of the previously enumerated conditions have tested false and the node is a Slave, then the packet is accepted and further processed where the DS sequence number of the packet is one more than the DS sequence number of the node. This signifies that the packet is the next in sequence. The processing of the packet in this situation is according to the two-step procedure as follows:

As a first step, the contents of the Database Summary packet which are acknowledged as having been received are then processed in the manner known in the existing PNNI protocol, as explained on pages 94 and 95 in Section 5.7.6 of the PNNI Specification under the heading "Exchanging". As a second step, the following actions are performed. The DS sequence number is set to the DS sequence number appearing in the received packet. Then, a Database Summary packet is sent to the Master. Next, if the received packet has its More Bit set to

zero, and the just transmitted Database Summary packet is empty and thus it also has its More bit set to zero; then the event (Re)Synch Done 25 is generated if the PTSE Request List is empty. If the PTSE Request List is not empty, the event Exchange (in Full) Done 26 is executed.

8) Where all of the previously enumerated conditions have tested false, the event DS Match is generated and processing of the Database Summary packet is stopped.

With regard to the Loading in Full State 24, a node in this state will have sent and received an entire sequence of Database Summary packets with the Sync in Full Bit set. Thus, the only Database Summary packets received by the node in this state should be duplicates. Any other Database Summary packets received that are not requesting a re-synchronization must generate the event DS Mismatch. This causes a reversion to the Negotiating State 14 and the two neighbouring peers in question will then proceed to synchronize their databases according to known procedures. The procedures to be followed when receiving a Database Summary packet in the Loading in Full State 24 are the same as those followed in the Exchange in Full State 22, except that packets accepted as the next in sequence must generate the event DS Mismatch instead, and further processing of such packets is stopped. Receipt of packets with an inconsistent Master Bit will also generate the event DS Mismatch. Any packets with the Initialize Bit as the only inconsistent bit will generate the event DS Resynch Mismatch.

As previously mentioned, an additional flag is defined according to the present invention for the otherwise known Database Summary packet of the existing PNNI protocol. For instance, bit number 13 in the known Flags field of the Database Summary packet may be used to define this new flag, which for convenience may be termed the Synch in Full (SF) Bit. This bit is set to a value of one by each of two nodes that will perform a database re-synchronization to distinguish this exchange process from synchronization as known in the existing PNNI protocol.

Those skilled in this art will appreciate that in some instances, networks having topology state routing protocols may be associated with other protocols which are pre-requisite to the procedures of database synchronization. For instance, in the specific case of the existing PNNI protocol, it must be ensured in applying the present invention that the Hello protocol is not disrupted by reason of the activity switch from the active routing entity to the inactive routing entity. If this were not the case, the network node associated with the failed active routing entity would be declared unreachable solely by reason of the activity switch. Thus, in the case of network protocols such as the PNNI protocol, local state information associated with the active routing entity must be provided to the inactive routing entity such that each will share a common understanding of local status information during recovery from failure. This may be accomplished, for instance, by periodic transmission of local state information from the active routing entity to the inactive routing entity prior to the failure in question.

The invention will next be described with reference to Figure 4 in relation to a hot redundant network element 56 in a communications network such as a network 2 (Figure 1) comprising network domains 30 and 55. The network element 56 may also implement a Logical Group Node, such as the node 36. The network element 56 comprises an input/output port 100, through which network connections 76 are routed by means of switching fabric 71. Network connections 76a may be re-routed or diverted as at 76b from an active routing entity 68 to an inactive routing entity 70 upon a failure of the active routing entity 68. Both the active routing entity 68 and the inactive routing entity 70 are provided with access to topology state information concerning the communications network 2 having network domains 30 and 55. For instance, the topology state information in question may be accessible to the active routing entity 68 and the inactive routing entity 70 by way of respective topology databases 74a and 74b.

According to the present invention, both the active routing entity 68 and the inactive routing entity 70 of a hot redundant network element such as node 56 may share

topology state information with each other by way of periodic transmissions 69 of topology state information from the active routing entity 68 to the inactive routing entity 70 which take place at predetermined intervals prior to the occurrence of a failure affecting the active routing entity 68. Alternatively, such transmissions may occur as new data is received by the active routing entity 68 in an event driven manner. As well, the inactive routing entity 70 may instead periodically be updated with topology state information supplied externally of node 56, for instance via externally sourced transmissions 69a. Those skilled in this art will appreciate that it is also possible to adapt the method of the present invention for use with a node 56 having an inactive routing entity 70 which is not periodically updated with topology state information. Those versed in this art will also understand that the sharing of topology information between the active and inactive routing entities 68, 70 of network element 56, as well as any other mechanism for periodically updating the topology state information of the inactive routing entity 70, may be conducted according to known techniques.

As is known in the art of hot redundancy techniques, not only may it be desired to update periodically the topology state information of the inactive routing entity 70, but in like manner, local state information may also be periodically transmitted to the inactive routing entity 70. For example, in the PNNI protocol, such local state information is transmitted by means of the Hello protocol. As previously explained, this is to prevent that network nodes do not declare a failed node unreachable for purposes of routing solely by reason of the activity switch which created the need for a database resynchronization. The recovered node will therefore retain its local state information following the activity switch in question. Such local state information would be transmitted in an analogous manner to that previously described in relation to topology state information.

When a node 56 (and logical node 36 which it implements) has previously synchronized topology databases with its neighbouring peers and would like to re-synchronize these databases, whether to recover from a failure local to the active routing

entity 68 or to perform a reset thereof, the implementation of the present invention will permit the node to initiate a topology database exchange process with its neighbour nodes which does not result in termination of the advertisement of local state information as it pertains to the node associated with the failure of the active routing entity 68 and as it pertains to each node which is immediately adjacent to the node associated with the failure of active routing entity 68. Such immediately adjacent nodes, as the terminology is used herein, include both physical and logical neighbour nodes. For instance, in the PNNI protocol, link advertisements among the various nodes involved in recovery from the failure of active routing entity 68 will not be terminated during database exchange. Thus, according to the present invention, the node associated with the failure of active routing entity 68 will not withdraw the advertisement of its local state information. As well, according to the present invention, the neighbour nodes which are immediately adjacent the node associated with the failure of active routing entity 68 likewise will not withdraw advertisement of the respective local state information pertaining to each such neighbour node.

The method of the present invention as described above may be implemented in each of the active routing entity 68 and the inactive routing entity 70 by respective database synchronization processors 78a and 78b, which may be realized in hardware, software or a combination thereof. Synchronization processors 78a and 78b perform the various procedures previously described in relation to the exchange of topology state information between the node 56, each of its immediately adjacent lower-level neighbours 57, 59, 60 and between node 36 and each of its immediately adjacent higher-level logical neighbours 34, 35, 37, 38. Although what has been described above provisions a respective synchronization processor 78a or 78b with each of the active routing entity 68 and the inactive routing entity 70, those skilled in this art will appreciate that a single synchronization processor may also be shared by both of the devices.

Thus, the present invention defines a database re-synchronization mechanism between a node and immediately adjacent physical or logical neighbour nodes thereto which causes these nodes not to alter their advertisements of local state information during the synchronization process. This should ensure that the synchronization between the nodes in question will be transparent to all other nodes in the network, namely those nodes which are not the immediately adjacent physical or logical neighbouring nodes of the failed node. The latter neighbouring nodes are the only nodes which need be called upon to take part in re-synchronization.

According to the present invention, an attempt is therefore made to address certain of the shortcomings of known prior art systems for database resynchronization, as may occur during hot redundancy recovery. In cases where the failed node is an active routing entity possessing distinguished responsibilities, the failure of the active routing entity is not as disruptive of the routing system or the network in general in the sense that none of the distinguished responsibilities of the routing entity need be relinquished during recovery. Hence, no other node need assume the distinguished responsibilities of the failed node. For instance, the Backup Designated Router of the OSPF protocol would not become the Designated Router and a new Peer Group Leader would not be elected at any level of the network hierarchy under the PNNI protocol as replacements for a failed distinguished network component.

Since fault detection and fault recovery in the present invention occur within the failed node and only with respect to physical or logical immediately contiguous nodes or devices, the recovery process according to the present invention is expected to be less time intensive than that of prior art techniques for hot redundancy recovery. As well, given that there is expected to be no involvement of network devices other than those described above which are contiguous to the failed network entity, a less extensive portion of the network topology participates in recovery than would be the case with the prior art techniques or devices. This expected advantage in the time and topological

extent of recovery is applicable whether or not the failed node is normally performing distinguished responsibilities. Lastly, network bandwidth and processing requirements associated with the recovery process of the present invention are expected to be reduced over known techniques of fault recovery in that only those nodes which immediately neighbour a failed component are involved in the recovery.

Although the present invention has been described with reference to an implementation adapted to the existing PNNI protocol, those skilled in the art of communications networks will appreciate that the invention may be applied or adapted to other topology state protocols. As well, while a synchronization processor may be employed for executing the method of the present invention, the various steps of the method may be performed in whole or in part by one or more other devices or software associated with a network element, or may be performed by existing devices or software which are associated therewith and which may be modified or otherwise adapted for use with this invention. Further, although the present invention has been described in the context of hot redundancy recovery, the invention may be employed in other situations of failure recovery or whenever it may be desirable for a node to resynchronize its topology database once it has previously performed a database synchronization. Moreover, those skilled in this art will understand that various modifications of detail may be made to the present invention, all of which come within its spirit and scope.

4. Brief Description of Drawings

FIGURE 1 is a schematic representation of a hierarchical network topology associated with a network domain operating according to the PNNI routing protocol in which the method and apparatus of the present invention may be implemented, and showing a parent-child relationship between groups of nodes forming part of the network topology.

FIGURE 2 is a state machine diagram which illustrates various states and transition events for a neighbouring peer Finite State Machine of the PNNI routing protocol as known in the prior art.

FIGURE 3 is a state machine diagram which illustrates various states and transition events for a neighbouring peer Finite State Machine of the PNNI routing protocol as modified to implement the present invention.

FIGURE 4 is a block diagram of a hot redundant network element in which the method of the present invention may be implemented.

Fig. 1

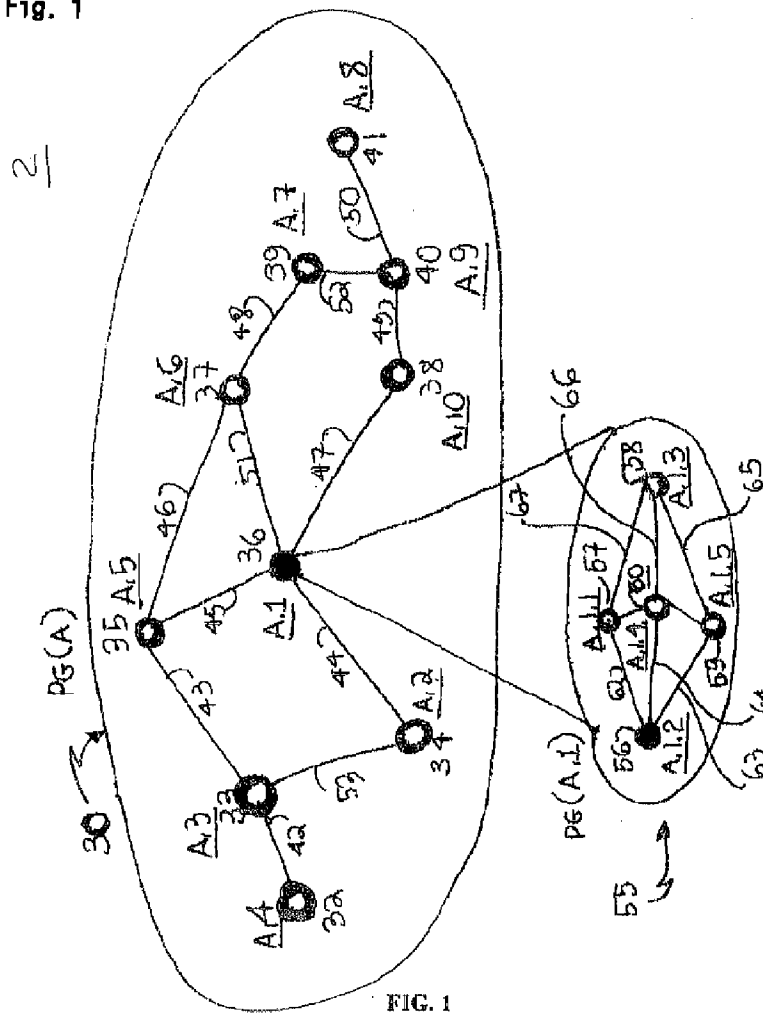


Fig. 2

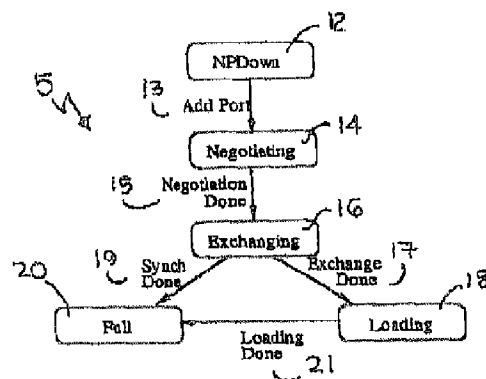


Fig. 3

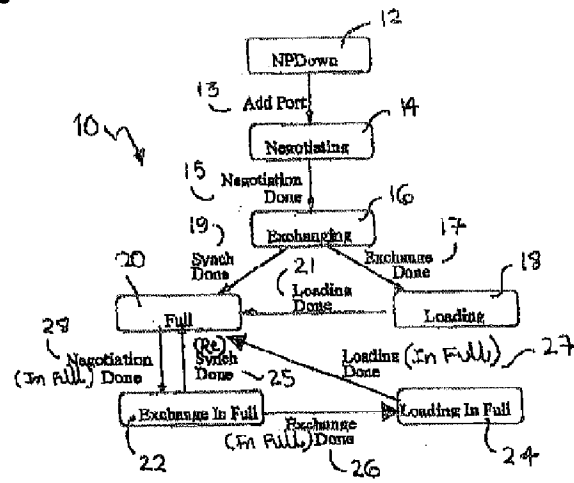


FIG. 3

Fig. 4

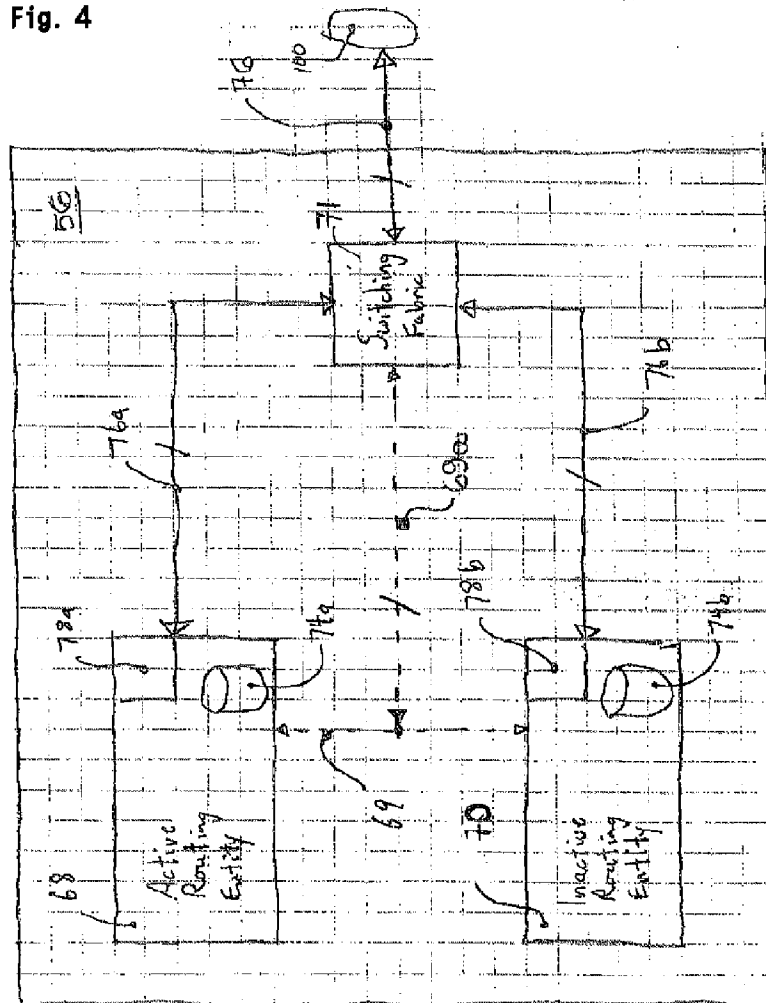


FIG. 4

1. Abstract

There is provided a method and apparatus for synchronization of topology state information between two network nodes in a communications network. The communications network has a routing protocol for intermittent advertisement of local state information throughout the network. The two network nodes include a requesting node which initiates a request for topology state synchronization and a replying node which receives the request. The replying node communicates with the requesting node to provide topology state information to the requesting node which is not possessed by the requesting node when the requesting node initiates its request. The method includes the step of selecting, prior to the request being made by the requesting node to the replying node, between a first and a second mode of synchronization. The first mode provides for topology state synchronization which entails withdrawal of the intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node. The second mode provides for topology state synchronization which maintains the intermittent advertisement of local state information as it pertains respectively to the requesting node and to the replying node.

2. Representative Drawing

Fig. 1